

**Oracle® Communications Billing and Revenue
Management**

Elastic Charging Engine 11.3 Concepts

Release 7.5

E70766-03

December 2016

Oracle Communications Billing and Revenue Management Elastic Charging Engine 11.3 Concepts, Release 7.5

E70766-03

Copyright © 2013, 2016, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	vii
Audience	vii
Documentation Accessibility	vii
Accessing Oracle Communications Documentation	vii
Document Revision History	viii
1 Overview of Elastic Charging Engine	
About Elastic Charging Engine	1-1
How ECE Charges Usage Requests for Online Network Mediation System	1-2
How ECE Charges Usage Requests for Offline Network Mediation System	1-3
About ECE and PDC	1-4
About ECE and BRM	1-5
About ECE and Policy Clients	1-5
How ECE Processes Policy Requests for Online Network Mediation Software	1-6
Integrating Charging Applications with ECE	1-8
2 ECE System Architecture	
Overview of ECE System Architecture	2-1
About Core ECE Components in the Cluster	2-3
About Diameter Gateway	2-4
About RADIUS Gateway	2-4
About Elastic Charging Server	2-5
About High Availability with Elastic Charging Server	2-5
About the Elastic Charging Client	2-6
About Elastic Charging Controller	2-6
About configLoader	2-7
About Pricing Updater	2-7
About Customer Updater	2-8
About External Manager Gateway	2-9
About BRM Gateway	2-9
About Rated Event Publisher	2-10
About Rated Event Formatter	2-10
About ECE as a Subscriber Profile Repository	2-11
About ECE Request Types	2-11

3 Understanding Charging Scenarios

Charging Overview	3-1
About Online Charging Scenarios	3-1
About Offline Charging Scenarios	3-2
About Synchronizing BRM and ECE Customer Data	3-2
About Usage-Charging Attributes	3-2
About Pricing Components	3-3
About Rating	3-3
About Rerating	3-4
About Advice of Charge	3-5
About Advice of Promotion	3-5
About Multiple-Service Credit Control	3-6
About Event-Based Charging	3-7
About Session-Based Charging	3-7
About Midsession Rated Events	3-8
About Discounts	3-8
About Balance Management	3-9
About Charge Distribution	3-9
About Reservation Validity	3-9
About First-Usage Validity	3-10
Granting First Usage Resources During Rating	3-10
About Reverse Rating	3-10
About Policy-Driven Charging	3-11
About Subscription Products and Member Products	3-13
Event Charging with Unit Reservation	3-13
Session Charging with Unit Reservation	3-13
Immediate Event Charging	3-13
Server Initiated Reauthorization Requests	3-13
About Taxation Support	3-14
About Using Closed User Group Profiles to Determine a Price	3-14
About Assigning Balance Impacts to Bill Items	3-15
About Processing Delayed Usage Requests	3-15
About Redirecting Subscriber Sessions to a Service Portal	3-16

4 About Usage Requests

Usage Request Overview	4-1
How Network Mediation Systems Create and Submit Usage Requests	4-2

5 About Usage Responses

Understanding Usage Responses	5-1
-------------------------------------	-----

6 About Notifications

Overview of Notifications in ECE	6-1
About In-Session Notifications	6-1
About External Notifications	6-2
About Notifications Used by BRM	6-3

About Notifications Used by Policy Clients	6-4
Spending Limit Notifications	6-4
Subscriber Preference Notifications	6-5
About Service Events that Trigger External Notifications	6-5
About Top-Up Notifications	6-5
About Billing Notifications	6-5
About Breach-Type Notifications	6-6
About Threshold Breach Notifications	6-6
About Credit Ceiling Breach Notifications	6-6
About Credit Floor Breach Notifications	6-6
About the Notification Framework	6-6
Transforming the Service Event Java Payload into XML	6-7

A Specifications and Standards Compliance in ECE

About Specifications and Standards Compliance	A-1
---	-----

Glossary

Preface

This guide provides an overview of Oracle Communications Billing and Revenue Management (BRM) Elastic Charging Engine (ECE).

Audience

This guide is intended for a wide reading audience, including:

- Business analysts who want to understand the charging capabilities of ECE as a rating engine.
- Developers who configure and extend the ECE software to support business requirements. For example, a charging domain expert who makes decisions about how network events are charged.
- Application administrators who configure ECE to support company product offerings and need to know how ECE processes requests for charging from the network.
- Solution architects or designers who need to know how ECE works within a broader charging system.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Accessing Oracle Communications Documentation

ECE documentation and additional Oracle documentation; such as Oracle Database documentation, is available from Oracle Help Center:

- <http://docs.oracle.com>

Additional Oracle Communications documentation is available from the Oracle software delivery website:

- <https://edelivery.oracle.com>

Document Revision History

The following table lists the revision history for this book.

Version	Date	Description
E70766-01	April 2016	Initial release.
E70766-02	September 2016	Documentation updates for ECE 11.3 Patch Set 1. <ul style="list-style-type: none">■ Updated the following sections:<ul style="list-style-type: none">About Rated Event FormatterAbout Notifications Used by BRMAbout In-Session Notifications
E70766-03	December 2016	Documentation updates for ECE 11.3 Patch Set 2. <ul style="list-style-type: none">■ Added the following sections:<ul style="list-style-type: none">About Processing Delayed Usage Requestsdelayed usage request

Overview of Elastic Charging Engine

This chapter is an overview of Oracle Communications Billing and Revenue Management (BRM) Elastic Charging Engine (ECE).

About Elastic Charging Engine

Elastic Charging Engine is the core charging-engine technology of the BRM system. ECE is the single charging engine for both offline and online charging. Built on Oracle Coherence, ECE is scalable and resilient, and as its name suggests, ECE can scale (like an elastic band can stretch) when tasked with processing thousands of transactions per second. The ability of ECE to scale with its in-memory charging technology supports low service latencies and high performance. See "[ECE System Architecture](#)" for more information about ECE and Oracle Coherence.

You can use ECE to charge customers for their use of any product on any network, using any service, and using any payment type. For a convergent charging system, ECE can perform both online charging and offline charging when rating events from the network.

In the BRM system, ECE performs usage charging. ECE rates events and calculates charges for services such as telephony usage and content downloads. Specifically, ECE does the following:

1. Receives event information from a network mediation system.

For example, ECE receives event information from the ECE Diameter Gateway (the supported Diameter interface) or a third-party network mediation system.

2. Measures the event.
3. Applies a charge to the resulting measurement.
4. Adds the charge to the customer's account balance.

To receive event data, ECE processes *usage requests* that are created and submitted by online and offline network mediation software programs. Usage requests contain event information that is used for online charging and offline charging of network usage, such as the duration of a call or the amount of data that is downloaded.

To rate event data, ECE uses pricing data. ECE uses pricing data from Pricing Design Center (PDC), which defines rates and rating rules that are used to determine the price of an event.

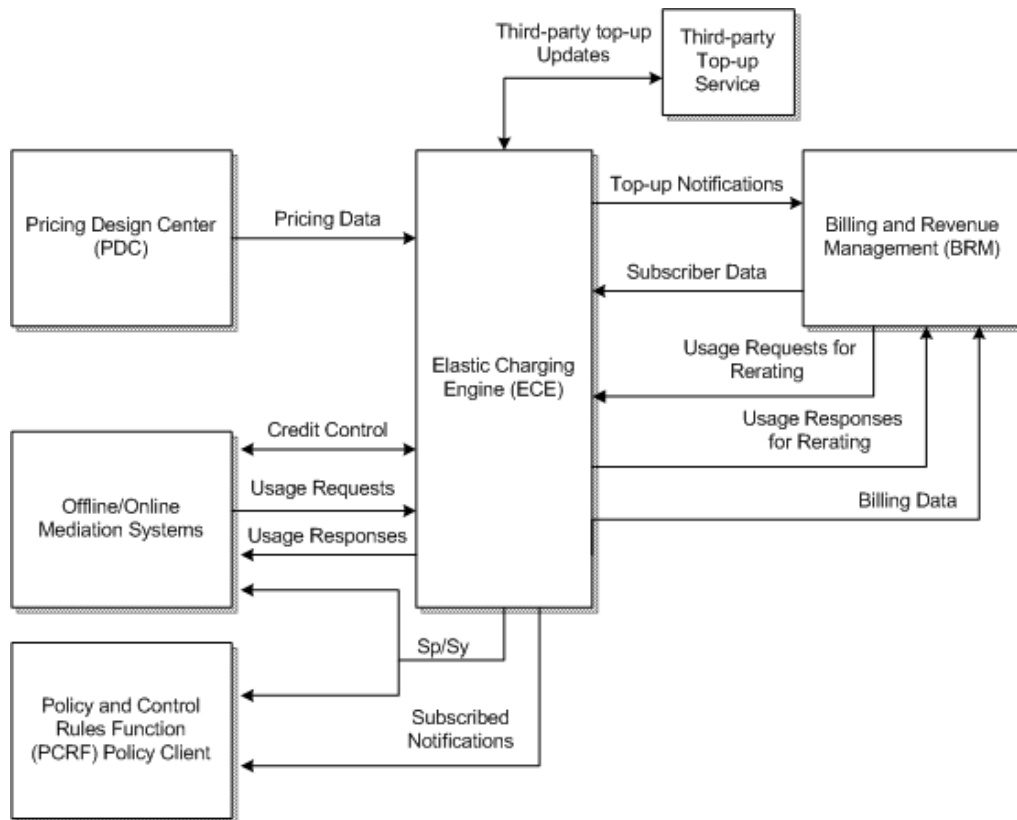
To rate event data, ECE also uses customer data. ECE uses customer data from BRM, which defines the products that the customer owns.

ECE performs real-time balance management, calculating how much quota a customer can use based on his balance. For example, ECE calculates how many minutes a caller can use based on his balance.

ECE has APIs for integrating with other systems in an integrated charging solution such as with the Policy and Charging Rules Function (PCRF) for implementing policy-driven charging and with top-up systems for applying top-ups to customer balances.

Figure 1-1 shows how ECE receives inputs from and provides outputs to other systems in an integrated charging solution.

Figure 1-1 ECE Inputs and Outputs To and From Other Systems in an Integrated Charging Solution



ECE can publish notifications when specific events occur in the ECE charging servers. Other applications can subscribe for receiving the notifications to use the data for their own processing. ECE uses notifications, for example, to provide network mediation systems with balance information for prepaid authorizations. ECE also uses notifications to provide balance updates to BRM.

In addition to usage requests, ECE processes various other types of requests. See ["About ECE Request Types"](#) for information.

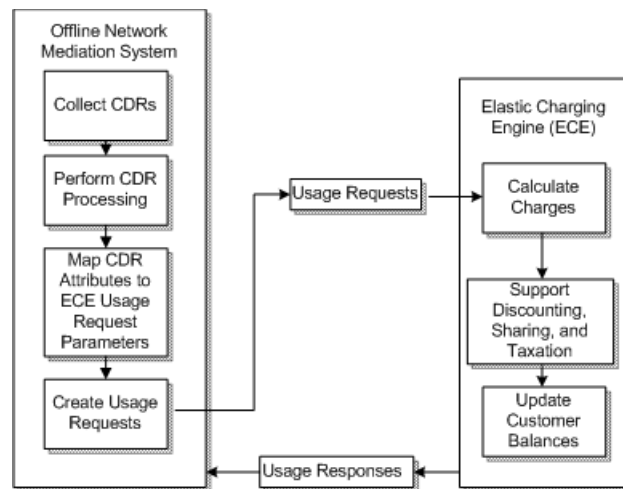
How ECE Charges Usage Requests for Online Network Mediation System

The following procedure describes how ECE typically processes a usage request for online charging.

3. The offline network mediation system transforms the CDRs into ECE usage requests.
The offline network mediation system extracts the event attributes and maps them to ECE usage request parameters.
4. The offline network mediation system submits the usage requests to ECE.
5. ECE calculates the charges based on the request attributes. ECE supports debit and refund operations for which the charges are passed in the input.
If ECE detects errors (for example, it does not find a valid offering for charging for the request), then it rejects the request. It sends a negative usage response to the offline mediation system which then rejects the CDR.
6. ECE manages the current state of customer balances within the charging system so that it is synchronized with the balance information in BRM.

Figure 1–3 describes how the offline network mediation system interacts with ECE.

Figure 1–3 Offline Mediation System and ECE



About ECE and PDC

ECE must have access to pricing data to rate the events contained in the usage requests it receives. ECE uses pricing data and pricing-related configuration data defined in PDC.

You design *pricing components* in PDC such as charge offers and discount offers. You also design *setup components* that support the creation of pricing components such as balance elements, impact categories, time models, rateable usage metrics (RUMs), service-event maps, and zone models. ECE uses the criteria in your pricing components to calculate the price of your services.

In PDC, you set up your *event definitions* for the events that you created in BRM. Your event definitions contains the request specification data that ECE client applications use for sending requests to the ECE charging servers. Your event definition includes the definition of usage attributes that ECE can accept when processing usage requests. You perform network enrichment of the event definition for your events in PDC. You add network attributes for all event attributes in the event definition that apply to usage-request charging operations. For information about event definition and mapping event attributes to network attributes, see *PDC User's Guide*.

See "[About Pricing Updater](#)" for a discussion on how PDC data is sent to ECE.

About ECE and BRM

After rating an event, ECE needs to send that rated event to BRM so that BRM can update the customer balance (debit or credit the customer balance) so the customer can be billed.

ECE first publishes the rated event to the Oracle NoSQL database data store that you previously configured for use by ECE. From the Oracle NoSQL database data store, the data is subsequently:

- **Extracted**

The Rated Event Formatter reads and extracts the rated event information from the Oracle NoSQL database data store so that it can be formatted.
- **Formatted**

Rated Event Formatter uses its `BrmCdrPlugindDirect` Plug-in to format the rated event information into CDR files in BRM CDR format.
- **Loaded into the BRM database**

The BRM Rated Event (RE) Loader subsequently loads the CDRs into the BRM database which updates the customer account balances in BRM.

BRM sends data to ECE when updates to customer data occur in the BRM system that must be pushed to ECE. See "[About Customer Updater](#)" for a discussion about how BRM data is sent to ECE.

BRM sends rerating requests to ECE (usage requests and usage responses that are sent for rerating). See "[About Rerating](#)" for a discussion about how BRM data is sent to ECE for rerating.

About ECE and Policy Clients

To support policy-driven charging, ECE offers a policy management API. Policy clients, such as Diameter Gateway, can use the API to retrieve data from the ECE data grid that is relevant for policy enforcement.

Policy-driven charging in ECE is based on the Policy and Charging Rules Function (PCRF), defined in the 3GPP TS 23.203 v9.9.0 (2011-06) specification. The PCRF can integrate with ECE through your online network mediation software.

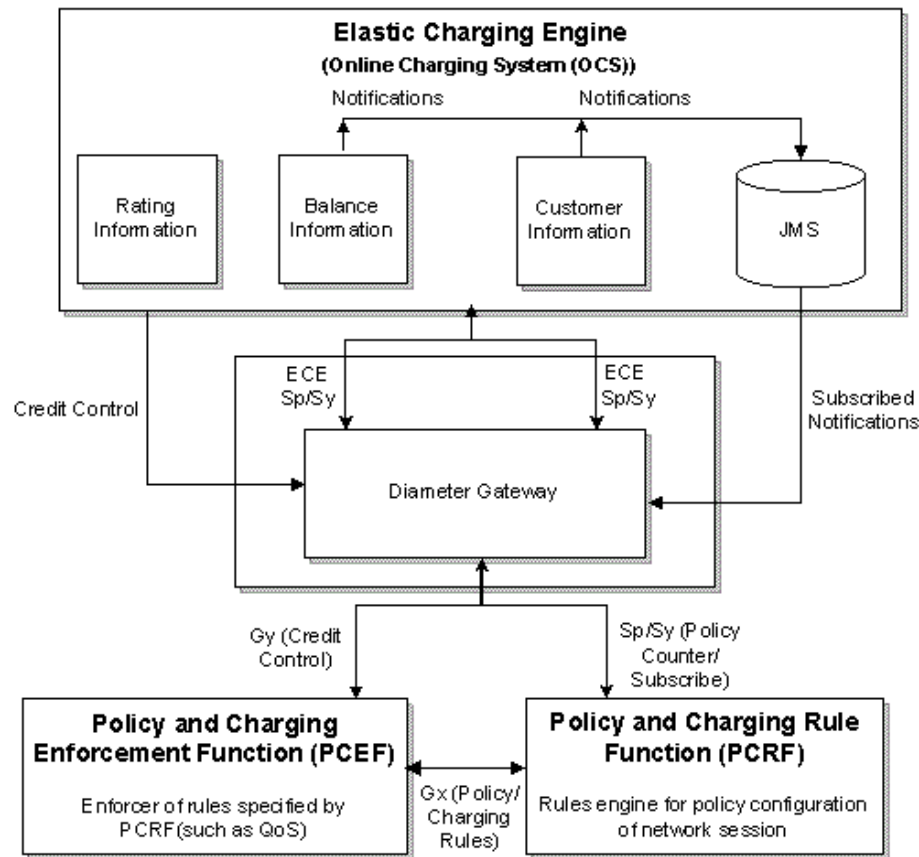
ECE exposes its cached in-memory data so that your online network mediation software can retrieve policy counter information and policy-related subscriber preference information. ECE publishes notifications containing the policy information and your online network mediation software uses the notifications to send the information to the PCRF for evaluation. For information about how ECE interacts with your online network mediation software for policy-driven charging, see "[How ECE Processes Policy Requests for Online Network Mediation Software](#)".

For more information about how ECE supports policy-driven charging, see "[About Policy-Driven Charging](#)".

For information about the ECE policy management API and the XML structure of ECE policy notifications, see the discussion about integrating policy clients with ECE in *BRM Elastic Charging Engine Implementation Guide*.

[Figure 1–4](#) illustrates how ECE fits into a charging system that implements policy-driven charging.

Figure 1–4 ECE and Policy Client Integration



How ECE Processes Policy Requests for Online Network Mediation Software

The following procedure describes how ECE processes requests for policy-driven charging from your online network mediation software (or from Diameter Gateway).

1. A customer starts to use a service which starts a network session.
For example, the customer turns on a mobile phone which connects to a wireless network.
2. At the start of the network session, the Policy and Charging Enforcement Function (PCEF) obtains from the Policy and Charging Rules Function (PCRF) a policy configuration.
The PCEF uses the Gx interface to obtain the policy configuration for the network session.
3. The PCRF requests the policy counters and subscriber preferences from your online network mediation software (or Diameter Gateway).
The PCRF uses the Diameter Sy/Sp interface.
4. Your online network mediation software initiates a policy session with ECE which does the following:
 - Requests policy counter and status label information.

Requests the policy counters for a specific product and subscribes for receiving notifications when the values of the policy counter information changes.

- Requests policy-related subscriber preferences information by doing *one* of the following:
 - Retrieves the value for a specified set of subscriber preferences and subscribes for receiving notifications when the values of the preferences change during the policy session.
 - Retrieves only the values for a specified set of subscriber preferences and does *not* subscribe for receiving notifications when the values of the preferences change during the policy session.

Your online network mediation software (and also Diameter Gateway) uses the PolicySessionRequest ECE Java combined Sy/Sp (implemented as Sh) interface which uses the SubscribeNotificationRequest procedure and the UserDataRequest procedure.

5. ECE sends a policy response to your online network mediation software (or to Diameter Gateway) which does the following:
 - Indicates whether the request succeeded or failed and a list of reasons supporting the response.
 - Sends the status of the policy counters as follows for the product specified, or if the product types were not specified, returns the information for all products:
 - Sends the offer profile name configured for the product.
 - Sends the status label associated with the policy counter.
 - Sends an *effective time* for the values of the policy counters. After the effective time expires, the PCRF is expected to send another request for policy counter and status label information (send another SpendingLimitReportRequest).
 - Sends the label name of the next probable status that will apply after the effective time expires. For example, **Medium_QoS**.
 - Sends the *delay interval*. The PCRF can use the delay interval and the effective time to determine when to query for the policy counters again.

ECE uses the SpendingLimitReportResponse procedure of the ECE Java Sy interface.

The information listed above is only a subset of the response. For all information included in the response, see *BRM Elastic Charging Engine Java API Reference*.

- Sends the subscriber preferences for the specified set of subscriber preferences. ECE uses the SubscribeNotificationResponse procedure of the ECE Java Sp interface.
6. The PCRF rules engine interprets the information and installs a policy on the PCEF which the PCEF enforces.
 7. A charging session is established and the PCEF sends a Ro message to your online network mediation software (or Diameter Gateway).
 8. Your online network mediation software (or Diameter Gateway) initiates a charging session with ECE.

9. ECE publishes policy notifications for the following:
 - Changes to the policy counter status for the policy counters the PCRF subscribed for (Sy data) at the beginning of the policy session.
 - Changes to the subscriber preferences the PCRF subscribed for (Sp data), if any, at the beginning of the policy session.

For information about policy notifications, see ["About Notifications Used by Policy Clients"](#).

10. Your online network mediation software (or Diameter Gateway) consumes the policy notifications and sends the data to the PCRF.

For information about policy notification data format, see the discussion about integrating ECE with policy clients in *BRM Elastic Charging Engine Implementation Guide*.

11. As the charging session continues, ECE performs credit control functions: rates events, authorizes usage events only if adequate balance is available, administers threshold checks based on the current balance and consumed reservation of the customer balance.

For information about how ECE processes usage requests, see ["How ECE Charges Usage Requests for Online Network Mediation System"](#).

12. When ECE detects a policy threshold breach during the charging session, it publishes a policy notification to the JMS notification queue containing the new status of the policy counter. Your online network mediation software (or Diameter Gateway) sends the data to the PCRF.

The customer balance change that causes the policy threshold breach could occur because of any of the following:

- Usage requests coming from the network mediation system
 - Update requests coming from BRM (a subscription activity in the customer management system)
 - Top ups coming from top-up systems
13. The PCRF evaluates the new policy counter values and the associated policy status labels and installs a new policy configuration on the PCEF.

The new policy is established dynamically during the charging session.
 14. The customer stops using his service which ends the network session.
 15. Your online network mediation software (or Diameter Gateway) terminates the charging session with ECE.
 16. Your online network mediation software (or Diameter Gateway) terminates the policy session with ECE.

Integrating Charging Applications with ECE

To integrate client applications such as mediation systems, Policy and Charging Rules Function (PCRF) clients, and top-up systems with ECE, you use the ECE APIs. See the Elastic Charging Engine API Overview in *BRM Elastic Charging Engine Implementation Guide* for a summary of the ECE APIs. Refer to *BRM Elastic Charging Engine Java API Reference* for detailed information about each type of ECE API.

For sending usage requests to ECE, client applications must call the ECE charging APIs according to the usage request builder defined. See the discussion of usage

requests in *BRM Elastic Charging Engine Implementation Guide* for information about how usage requests are built and the supported operation types.

An ECE SDK is included with the software and contains sample programs that demonstrate the arguments and methods programs must use when calling the ECE APIs. See the discussion of using ECE testing tools in *BRM Elastic Charging Engine Implementation Guide* for information about the sample programs.

When you introduce different ways of charging customers for the use of your products, you can extend existing usage requests or create new usage requests to support the attributes required for the different ways of charging. When extending or creating usage requests to handle new charging attributes, you must update integration points in the charging system so that products that surround ECE can handle the new attributes.

In addition to integrating client applications, you also need to format data from these applications to conform to the ECE data schema files. See the discussion of loading data in *BRM Elastic Charging Engine Implementation Guide*.

ECE System Architecture

This chapter describes the Oracle Communications Billing and Revenue Management Elastic Charging Engine (ECE) system architecture.

Overview of ECE System Architecture

ECE is a grid application built on Oracle Coherence. A grid application is an application that leverages a *data grid*: a system composed of servers that work to manage information and related operations in a distributed environment. ECE implements distributed functions over a network of nodes. These nodes are Java applications that include Oracle Coherence libraries. Services are leveraged to implement application functions in the ECE components. The main components are:

- **Elastic Charging Server**

Elastic Charging Server (ECE charging server nodes) receives and calculates data from ECE clients (client applications). Elastic Charging Server carries out all charging business logic, such as usage charging, and performs query and update operations on ECE cache data.

Elastic Charging Server sends information to external systems.

To send requests from the network to Elastic Charging Server for processing, ECE clients must join the ECE cluster to access the ECE data and other services required for sending requests. ECE clients that perform usage metering create *usage requests* that carry usage information to Elastic Charging Server to calculate usage. Client applications use the Elastic Charging Client (described below) to join the ECE cluster.

Elastic Charging Server supports *query requests* from external systems, such as customer self-care or network mediation systems, so these systems can obtain data from ECE caches. For self-care systems, query requests are sent to Elastic Charging Server when customers query their account balance and account password information. ECE clients use the Elastic Charging Client (described below) to send query requests.

See "[About Elastic Charging Server](#)" for more information.

- **Elastic Charging Client**

The Elastic Charging Client is a client library installed on ECE clients that enables ECE clients to connect and send requests to ECE. When ECE clients start the Elastic Charging Client, they automatically join the cluster that provides access to ECE node data caches and the services required for sending requests. The Elastic Charging Client is used for sending all types of requests: usage, query, update, and management.

See "[About the Elastic Charging Client](#)" for more information.

- **Diameter Gateway**

Diameter Gateway translates Diameter requests received from Diameter clients (for example, application servers, policy servers, or IP multimedia subsystem gateway functions) into ECE Java API requests. It translates the response from Elastic Charging Server back into Diameter requests and responds back to the requesting Diameter client.

The Elastic Charging Client is integrated in Diameter Gateway.

See "[About Diameter Gateway](#)" for more information.

- **RADIUS Gateway**

RADIUS Gateway translates RADIUS requests received from RADIUS clients into ECE Java API requests. It translates the response from Elastic Charging Server back into RADIUS requests and responds to the requesting RADIUS client.

The Elastic Charging Client is integrated in RADIUS Gateway.

See "[About RADIUS Gateway](#)" for more information.

- **Customer Updater and Pricing Updater**

To ensure that ECE has the most current data to use in charging calculations:

- Customer Updater manages customer data update requests from the BRM server to ensure that customer data and balances are current in the ECE cache. See "[About Customer Updater](#)".
- Pricing Updater processes update requests from Pricing Design Center (PDC) to ensure that pricing data is current in the ECE cache. See "[About Pricing Updater](#)".

- **BRM Gateway**

BRM Gateway synchronizes Oracle Communications Billing and Revenue Management BRM data with ECE data by sending update requests from ECE to BRM. ECE sends update requests to BRM for various reasons: for example, to BRM to update the subscriber life-cycle state in BRM and to trigger billing in BRM for a customer. See "[About BRM Gateway](#)" for more information.

- **Rated Event Publisher and Rated Event Formatter**

Rated Event Publisher and Rated Event Formatter persist ECE-generated rated events and send them to external systems. ECE publishes rated events to the Oracle NoSQL database by using Rated Event Publisher. ECE formats rated events for the BRM system by using Rated Event Formatter. For a non-BRM system, rated events are formatted via a custom plug-in. See "[About Rated Event Publisher](#)" and "[About Rated Event Formatter](#)" for more information.

- **Elastic Charging Controller and Oracle Enterprise Manager Cloud Control**

Elastic Charging Controller (ECC) and Oracle Enterprise Manager Cloud Control monitor and maintain the ECE system. You install these applications on the driver machine, which is the server that is set up to be the primary administrator machine.

A system administrator who wants to maintain, update, and tune the ECE system uses ECC. ECC is the ECE command-line application used for day-to-day administration, managing, and operating of the ECE system. See the discussion of using ECC in *BRM Elastic Charging Engine System Administrator's Guide* for more information.

A systems operator who wants to monitor the ECE system and perform basic operational tasks uses Oracle Enterprise Manager Cloud Control. ECE leverages Oracle Enterprise Manager Cloud Control for operations management. To use Oracle Enterprise Manager Cloud Control for ECE, you must install Oracle Application Management Pack for Oracle Communications. Oracle Application Management Pack for Oracle Communications provides management capabilities for BRM and other supported Oracle Communications applications. For detailed information about the management capabilities provided by Oracle Application Management Pack for Oracle Communications, see *Oracle Application Management Pack for Oracle Communications System Administrator's Guide*.

For general information about monitoring and managing ECE components, see the discussion about monitoring and managing ECE in *ECE System Administrator's Guide*.

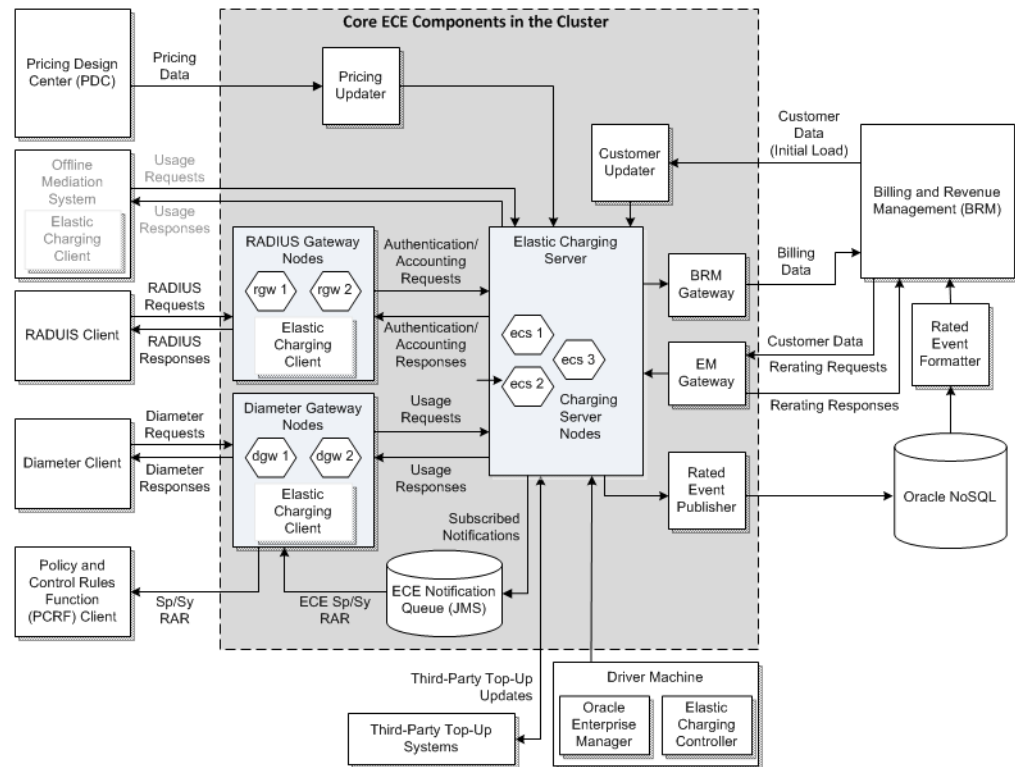
- **External Manager Gateway**

External Manager (EM) Gateway enables BRM to send requests to and receive responses from ECE by converting BRM opcode flists into ECE Java API-formatted requests. BRM sends requests to ECE for various reasons. For example, during rerating, BRM requests the calculation of usage charges for events that need to be rerated. See "[About External Manager Gateway](#)" for more information.

About Core ECE Components in the Cluster

Figure 2-1 shows the core ECE components or nodes in the cluster. The core components are shown in the Core ECE Components in the Cluster box.

Figure 2-1 How Core ECE Components Interact with External Applications



About Diameter Gateway

Diameter Gateway translates Diameter requests received from Diameter clients (for example, application servers, policy servers, or IP multimedia subsystem gateway functions) into ECE Java API requests. It translates the response from Elastic Charging Server back into Diameter requests and responds back to the requesting Diameter client.

Diameter Gateway serves as the online charging system (OCS) front-end server for the BRM system. It processes network requests for Gy, Sy, and Sh Diameter interfaces. It acts as a Diameter server and presents ECE to the network as a Diameter credit-control application.

Diameter Gateway and Elastic Charging Server run on the same Coherence cluster, which provides high availability and service continuity.

Diameter Gateway communicates with Elastic Charging Server through ECE Java API calls.

Diameter Gateway exposes custom interfaces for top-up and balance query operations.

Diameter Gateway reads from the ECE notification queue (JMS topic) and processes push notifications from Elastic Charging Server. From the ECE push notifications (ECE re-authorization request (RAR) notifications, ECE spending limit notifications, and ECE subscriber preferences notifications), Diameter Gateway constructs Sy and Gy messages and sends them back to Diameter clients (for example, for answering back to push-notification-request (PNR), subscribe-notification-request (SNR), and RAR notifications).

For information about adding Diameter Gateway nodes to your topology, connecting node instances to the network, and configuring node instances, see *BRM Elastic Charging Engine System Administrator's Guide*.

For information about using Diameter Gateway for processing Gy, Sy, and Sh interface request types, see the discussion about network integration for online charging using Diameter Gateway in *BRM Elastic Charging Engine Implementation Guide*.

For information about the Diameter standard attribute-value pairs (AVPs) and Oracle custom AVPs used by Diameter Gateway for processing Gy, Sy, and Sh interface request types, see *BRM Elastic Charging Engine Diameter Gateway Protocol Implementation Conformance Statement*.

About RADIUS Gateway

Important: RADIUS Gateway is an optional component that requires a separate license.

RADIUS Gateway translates RADIUS requests received from RADIUS clients into ECE Java API requests. It translates the response from Elastic Charging Server back into RADIUS requests and responds to the requesting RADIUS client. You use RADIUS Gateway to process authentication and accounting requests when your customers use terminal servers or Network Access Server (NAS) to connect to ECE.

RADIUS Gateway serves as the OCS front-end server for the BRM system. It acts as a RADIUS server and presents ECE to the network as a RADIUS application.

RADIUS Gateway provides custom interfaces for processing authentication and accounting requests from RADIUS clients.

RADIUS Gateway is included in the ECE server installation, and you can deploy RADIUS Gateway nodes into the ECE cluster the same way you deploy other ECE nodes. RADIUS Gateway has ready-to-use example configuration files to facilitate implementation.

For information about adding RADIUS Gateway nodes to your topology and configuring node instances, see *BRM Elastic Charging Engine System Administrator's Guide*.

For information about mapping RADIUS network attributes to event attributes and customizing the RADIUS data dictionary in ECE, see the discussion about configuring and customizing RADIUS Gateway in *BRM Elastic Charging Engine Implementation Guide*.

For more information about using RADIUS Gateway for processing authentication requests, see the discussion about authentication using RADIUS Gateway in *BRM Elastic Charging Engine Implementation Guide*.

For more information about using RADIUS Gateway for processing accounting requests, see the discussion about accounting using RADIUS Gateway in *BRM Elastic Charging Engine Implementation Guide*.

For information about the custom interfaces and the extension points available for processing requests from RADIUS clients, see *BRM Elastic Charging Engine Extensions*.

For information about the RADIUS standard AVPs and messages used by RADIUS Gateway for processing requests from RADIUS clients, see *BRM Elastic Charging Engine RADIUS Gateway Protocol Implementation Conformance Statement*.

About Elastic Charging Server

Elastic Charging Server uses its charging server nodes to perform all charging business logic, such as usage charging, query, and update logic.

By using the Oracle Coherence in-memory data grid, ECE can distribute objects required for usage charging and their related processing across multiple physical servers, eliminating single points of failure and single points of bottleneck. In addition, Coherence enables vertical and horizontal scalability for ECE.

ECE charging servers receive usage requests containing usage information from the Elastic Charging Client. Customer and pricing data required to process ECE requests are cached within the ECE cluster (where transaction processing occurs) so the ECE charging servers are not required to perform data lookups in external applications.

About High Availability with Elastic Charging Server

Elastic Charging Server has built-in high availability features. ECE components that are part of the overall charging system, such as gateways, clients, and the Oracle NoSQL database, also have built-in high availability features.

ECE charging server nodes use Oracle Coherence technology and leverage features of Coherence clusters for high availability, including:

- Distribution of ECE objects across multiple physical servers
- Distributed processing, with no single points of failure or contention

- Storage of information in multiple nodes across multiple physical servers and racks to protect against node, machine, and rack failure

About the Elastic Charging Client

The Elastic Charging Client is a client library installed on ECE clients that enables ECE to connect and send requests to ECE. When ECE clients start the Elastic Charging Client, they automatically join the cluster that provides access to ECE node data caches and the services required for sending requests. The Elastic Charging Client is used for sending all types of requests: usage, query, update, and management.

Real-time charging events and offline call details records (CDRs) from the network are transformed by network mediation software programs into usage requests that use a format required by ECE. The network mediation software programs use the ECE charging API for creating the usage requests.

The API payload is extensible so that you can define charging attributes for creating usage requests that satisfy your business (and charging) requirements. These request specifications can be extended (or you can create new specifications) when you perform event definition in PDC. See "[About Usage Requests](#)" for information about the usage-request construction required by ECE.

The Elastic Charging Client sends asynchronous usage responses back to client applications that send usage requests. It can send requests as a batch to Elastic Charging Server. The Elastic Charging Client can also send batch requests that belong to a specific charging server node in the cluster.

The Elastic Charging Client knows in which charging server node on the cluster the customer data is cached; it leverages this information to route requests for processing so that the processing of the requests can occur where the data is located.

The Elastic Charging Client records latency tracking information for how requests were handled in the system.

Oracle Communications Offline Mediation Controller integrates with ECE to send usage requests for offline charging. The Elastic Charging Client is installed on the Offline Mediation Controller distribution cartridge. See *Oracle Communications Offline Mediation Controller Elastic Charging Engine Cartridge Pack User Guide* for instructions on how to connect Offline Mediation Controller to ECE and submit requests to ECE for processing.

About Elastic Charging Controller

Elastic Charging Controller is used for managing the ECE server processes or nodes in the ECE cluster. Elastic Charging Controller reads your topology file to know where all of the nodes of the cluster are located.

Elastic Charging Controller is installed on the driver machine.

The following are some examples of tasks you can perform using Elastic Charging Controller:

- Starting and stopping ECE nodes in the cluster
- Pushing configuration changes from the driver machine to all server machines across which the cluster is deployed
- Performing rolling upgrades of certain components into a running ECE system, such as changing JVM tuning properties, changing JMS configuration properties, or upgrading extension module customizations (changing extension JAR files)

For information about using Elastic Charging Controller, see the discussion about managing ECE nodes in *BRM Elastic Charging Engine System Administrator's Guide*.

About configLoader

Both RADIUS Gateway and Diameter Gateway must have access to mediation specification data to create authentication, accounting, or usage requests. **configLoader** is a utility that loads ECE with mediation specification data that you configure in your mediation specification file.

For information about editing the Diameter mediation specification file, see the discussion about network integration for online charging using Diameter Gateway in *BRM Elastic Charging Engine Implementation Guide*.

For information about editing the RADIUS mediation specification file, see the discussion about configuring and customizing RADIUS Gateway in *BRM Elastic Charging Engine Implementation Guide*.

About Pricing Updater

ECE must have access to pricing data to rate the usage requests it receives. ECE uses pricing data and pricing-related configuration data defined in PDC.

Pricing Updater loads pricing data into ECE automatically when you publish pricing data in PDC for ECE.

When you publish pricing data in PDC, Pricing Updater writes the pricing data to a JMS queue. PDC sends pricing data in XML form in the PDC format.

Pricing Updater listens on the queue. Pricing Updater dequeues the data from the JMS queue, transforms the XML into the format understood by ECE (in ECE XML data format), and loads the data into ECE.

Pricing Updater loads request specification data into ECE. Request specification data is required by ECE client applications for the creation of usage requests. PDC automatically generates the request specification data that is required in XML form in the PDC format. Similar to its handling of pricing data, Pricing Updater dequeues the request specification data from the JMS queue, transforms the XML into the format understood by ECE (in ECE XML data format), and loads the data into ECE.

You provide information required for configuring Pricing Updater when you install ECE. You can change this configuration if needed. See the discussion of implementing ECE with PDC in *BRM Elastic Charging Engine Implementation Guide* for information about configuring Pricing Updater.

Caution: The **pricingLoader** utility is used to load sample pricing data into ECE from XML files that are included in the sample pricing data directory (*ECE_home/occeserver/sample_data/pricing_data*, where *ECE_home* is the directory in which ECE is installed). Sample pricing data conflicts with pricing data created using PDC.

Do *not* use **pricingLoader** on a production deployment.

Do *not* run **pricingLoader** and Pricing Updater on the same ECE deployment.

About Customer Updater

Customer Updater does the following:

- Performs the initial extraction of customer data, credit profiles, offer profiles, configuration objects, and pricing cross-reference data from BRM and loads the data into ECE
- Handles asynchronous updates from BRM to ECE of the following data: rerating, account migration, discount, and product

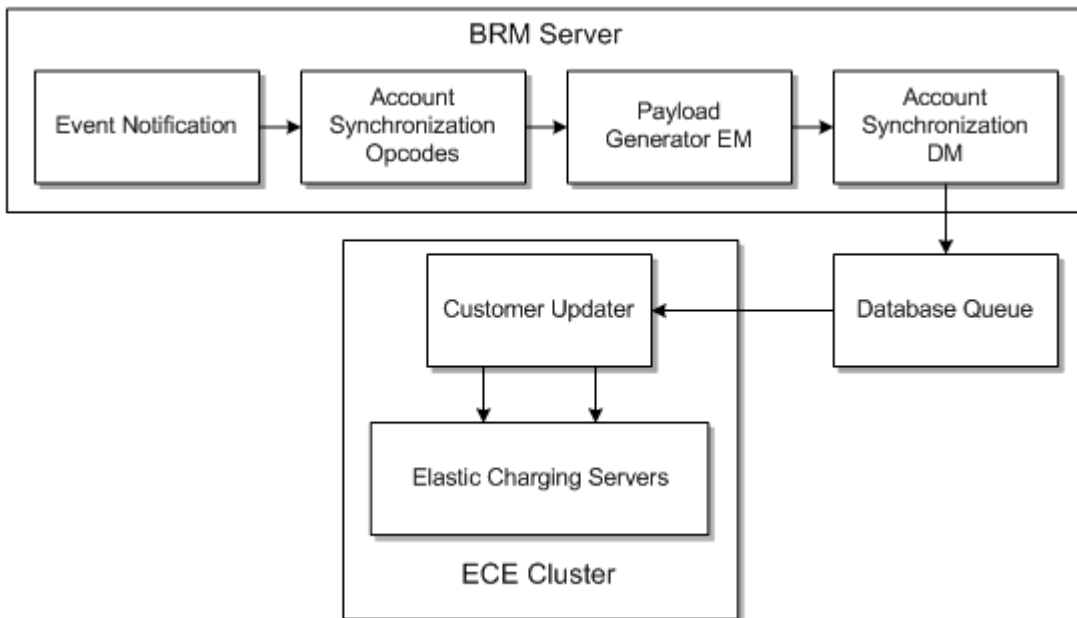
The rerating, account migration, discount, and product data stored in ECE must stay current with the corresponding data in the BRM database.

When events occur that update that data in the BRM database, the updates can be published asynchronously (not in real time) to ECE through Customer Updater. To do so, the updates are sent in business events to Account Synchronization Data Manager (DM), which publishes the business events to an Oracle Advanced Queuing (Oracle AQ) database queue. Customer Updater retrieves the business events from the queue and updates the ECE cache accordingly.

To ensure that the ECE cache is synchronized with the BRM database before usage processing begins, Customer Updater dequeues all requests from the database queue before putting ECE into a **usageProcessing** state.

Figure 2-2 shows the data flow from event notification to ECE update through Customer Updater:

Figure 2-2 Data Flow from Event Notification to ECE Update through Customer Updater



For general information about configuring event notification, account synchronization, and business event payloads, see the BRM documentation.

For information about configuring ECE event notification and payloads, see the discussion about loading the ECE configuration files into your BRM system in *BRM Setting Up Pricing and Rating*.

For information about configuring Customer Updater, see the discussion about implementing ECE with BRM in *BRM Elastic Charging Engine Implementation Guide*.

For information about updating *customer* data in real time, see "[About Synchronizing BRM and ECE Customer Data](#)".

Caution: Running the **customerLoader** utility without any parameter loads sample customer data into ECE from XML files included in the sample customer data directory (*ECE_home/occeserver/sample_data/customer_data*). Sample customer data conflicts with customer data created using BRM. Running the **customerLoader** utility with the **incremental** parameter loads customer data incrementally, in batches or in bulk, from BRM into ECE.

Do *not* run **customerLoader** without the **incremental** parameter on a production ECE deployment.

About External Manager Gateway

External Manager (EM) Gateway is a Java Virtual Machine (JVM) process that acts as a client to Elastic Charging Server, sending requests to ECE from BRM.

EM Gateway listens on a port for requests from BRM in opcode input flist format. EM Gateway converts the input flists into ECE requests and converts the subsequent ECE responses back into output flists.

The BRM server uses EM Gateway to publish *customer data updates* from the BRM database to the ECE cache. For more information, see "[About Synchronizing BRM and ECE Customer Data](#)".

The BRM rerating utility uses EM Gateway to send *rerating requests* from BRM to ECE during the rerating process. See "[About Rerating](#)" for more information.

Custom BRM applications can use EM Gateway to send *balance query requests* to ECE for querying the customer real-time usage balance that is managed in ECE.

For information about configuring EM Gateway, see the discussion about implementing ECE with BRM in *BRM Elastic Charging Engine Implementation Guide*.

About BRM Gateway

BRM Gateway handles update requests from ECE to BRM (ECE sends requests to BRM to update data in the BRM database).

BRM Gateway sends information to BRM when processes must be triggered in BRM or data must be updated in BRM due to ECE usage-request processing.

BRM Gateway connects to and sends information to BRM for the following purposes:

- Synchronize the first-usage validity initialized from ECE to BRM

See "[About First-Usage Validity](#)".

- Update the subscriber life-cycle state in BRM

When changes occur to a customer's subscriber life-cycle state in ECE, ECE publishes a service event. BRM Gateway uses the service event to send the subscriber life-cycle state information to BRM so that BRM can update the state in the BRM database.

For example, the subscriber life-cycle state can go from a pre-active state to an active state in ECE when a customer activates a prepaid calling card from the network. ECE uses BRM Gateway to send the new state information to BRM.

- Trigger billing in BRM

If a customer starts a session immediately before billing is set to run in BRM, ECE sends a request to BRM to run billing for that customer. See "[About Billing Notifications](#)" for more information.

- Refresh item POIDs in ECE

ECE gets a list of POIDs from BRM, caches them in ECE, and sets them in the rated event.

For tracking a customer's balance impacts for a given billable item type, ECE maps the product type and event type combinations in a usage request to its relevant item type. ECE sends information to BRM to refresh the item POIDs in the BRM database.

BRM Gateway is part of the ECE cluster, but it is not storage enabled; it is a Java process interacting with ECE through JMS. The primary purpose of BRM Gateway is to interact with BRM to send update requests.

BRM Gateway listens on the ECE notification queue for notification events that trigger update requests to BRM. When such a notification event occurs, BRM Gateway connects to the BRM Connection Manager (CM) and sends the relevant data from the notification event to BRM. In sending the information, BRM Gateway calls the necessary BRM opcodes that use the data for triggering processes in BRM.

You must configure BRM Gateway to connect to the BRM CM. See the discussion of configuring BRM Gateway in *BRM Elastic Charging Engine Implementation Guide*.

About Rated Event Publisher

Rated Event Publisher takes rated event information in Elastic Charging Server and publishes this information into the Oracle NoSQL database. Rated Event Publisher takes ECE Java objects in Elastic Charging Server and publishes them in (Coherence) binary format to a NoSQL datastore.

About Rated Event Formatter

Rated Event Formatter translates rated events into a format that can be loaded by Rated Event Loader. Rated Event Loader loads rated events into the BRM database. You can configure Rated Event Formatter to translate and send rated events in the required format to external systems, such as a data warehouse.

Rated Event Formatter runs plug-ins to translate rated events in different formats. The `BrmCdrPluginDirect` Plug-in transforms events from the Oracle NoSQL database into CDR files in the BRM CDR format. For a non-BRM system, you can configure a custom plug-in to transform rated events in different formats, such as XML and JSON.

When you use a custom plug-in, the custom plug-in and the `BrmCdrPluginDirect` Plug-in process the rated events simultaneously. When the rated events are processed by the non-BRM system and the BRM database, the rated events are purged from the BRM database.

For more information about the `BrmCdrPluginDirect` Plug-in, see the discussion about configuring the `BrmCdrPluginDirect` Plug-in in *BRM Elastic Charging Engine Implementation Guide*.

About ECE as a Subscriber Profile Repository

When integrating policy and charging rules function (PCRF) policy clients with ECE, ECE acts as the subscriber profile repository (SPR) because it stores the customer profile information used by the PCRF. ECE offers a combined Sp (implemented as Sh) and Sy interface, which the PCRF uses to retrieve customer preferences and policy counter information. For more information, see "[About ECE and Policy Clients](#)" and see the discussion about PCRF policy clients in *BRM Elastic Charging Engine Implementation Guide*.

About ECE Request Types

ECE receives various types of requests from other applications. ECE processes the following types of requests:

- Usage requests
- Policy requests
- Top-up requests (a type of update request)
- Query requests
- Management requests
- Update requests

ECE offers services that expose its client APIs that applications can use to create these requests. For more information, see the ECE API Reference appendix in *BRM Elastic Charging Engine Implementation Guide*.

Understanding Charging Scenarios

This chapter describes the charging scenarios supported by Oracle Communications Billing and Revenue Management Elastic Charging Engine (ECE).

Charging Overview

You can use ECE to perform online charging only, offline charging only, or both online charging and offline charging when used within a convergent charging system.

See the following topics for information about the charging scenarios ECE supports:

- [About Online Charging Scenarios](#)
- [About Offline Charging Scenarios](#)

ECE charges network events based on various attributes and the pricing components associated with them. See "[About Usage-Charging Attributes](#)" and "[About Pricing Components](#)".

ECE supports various rate types when charging network events. See "[About Rating](#)" for more information.

The ECE SDK includes sample programs that demonstrate how mediation system client applications call the ECE APIs for supported charging scenarios. See the discussion of using ECE developer tools in *BRM Elastic Charging Engine System Administrator's Guide* for information about the sample programs.

About Online Charging Scenarios

ECE supports the following charging scenarios for online charging of network events as defined by 3GPP standards:

- Event-based charging:
 - Event charging with unit reservation (ECUR)
 - Immediate event charging (IEC)
- Session-based charging:
 - Session charging with unit reservation (SCUR), including:
 - * Reverse rating
 - * Charge distribution
- Advice of Charge (AoC)

Additional capabilities include:

- Alterations (typically called *discounts* in Billing and Revenue Management (BRM))
- Distributions
- Advice of Promotion (AoP)
- Taxation
- Final Unit Indicator (can be used to support in-session top-ups)

ECE provides a Java interface for online charging of network events.

ECE provides a Java interface for supporting operations from policy and control network mechanisms (the policy and charging rules function (PCRF)). For the discussion of policy-related charging, see "[About Policy-Driven Charging](#)".

About Offline Charging Scenarios

ECE supports the following charging scenarios for offline charging of network events:

- Event-based charging

Additional capabilities include:

- Alterations
- Taxation

About Synchronizing BRM and ECE Customer Data

When the BRM server performs customer management and billing transactions, it stores the results in the BRM database. To enable ECE to rate usage events properly, all customer data updates made in the BRM database must also be made in the ECE cache.

The updates are applied to the ECE cache synchronously (in real time) as follows:

1. The BRM server performs the transaction and then sends the updates in business events to ECE through External Manager (EM) Gateway.
2. EM Gateway processes the business events and then informs the BRM server whether the ECE cache update was successful.
3. One of the following occurs:
 - If the cache update succeeds, the updates are saved to the BRM database.
 - If the cache update fails, the database updates are rolled back.

Because both the database and the cache are updated within the same transaction, no lag time occurs, and the BRM and ECE data remains synchronized whether the cache update succeeds or fails.

To enable synchronous customer data updates, see the discussion about enabling real-time synchronization of BRM and ECE customer data updates in *BRM Elastic Charging Engine Implementation Guide*.

About Usage-Charging Attributes

Attributes affect how ECE implements rating. Attributes can be associated with the event, the customer, or the product.

ECE can rate network events differently based on the following attributes:

- Chargeable-event-rating attributes
- Customer-rating attributes
- Product-rating attributes

For example, a customer-rating attribute can be a customer's birth date where you provide free text messages to customers on their birthday.

Chargeable-event-rating attributes must be defined in your event definition so that ECE identifies them as attributes that drive rating.

Customer and product attributes that drive rating in ECE at runtime are ECE customer model attributes. These attributes are defined in Oracle Communications Pricing Design Center (PDC) when you perform event definition in PDC so that you can define pricing rules based on them.

About Pricing Components

Pricing components are the rates that ECE uses for charging. Pricing components are defined in PDC. ECE uses the pricing components to configure the value of a chargeable event received from the network. See the PDC Help for instructions on how to create pricing components for ECE.

ECE supports Advice of Promotion as a system-level configuration; it is not configurable in PDC. See ["About Advice of Promotion"](#) for more information.

About Rating

To calculate charges and discounts during rating at runtime, ECE uses configuration data and pricing component data defined in PDC. ECE can use attributes from the chargeable event received from the network, attributes from the customer data received from BRM, and attributes from the product data received from BRM for determining what rate to apply.

ECE supports various rates for determining the value of a chargeable event. You define rates in PDC, including the following:

- Zero rate
- Flat rate
- Tiered rate
- Threshold rate
- Zone-based rate
- Time-based rate
- Day-based rate
- Special day-based rate

See the PDC Help for descriptions of the preceding rate types and how to define them in your ECE pricing components.

At runtime, ECE can perform reverse rating for session-based charging. See ["About Reverse Rating"](#) for more information.

About Rerating

You can rerate ECE-rated events if required. Rerating events can be required for various reasons. For example, if one of your existing charge offers was replaced between the last and next billing cycles.

See the discussion about rerating in the BRM documentation for information about rerating in the BRM charging system.

Rerating is initiated in BRM when you run the BRM **pin_rerate** utility. The **pin_rerate** utility sends rerate jobs to ECE for events that were originally rated by ECE.

ECE supports *concurrent rating* in which ECE can continue to rate a customer's real-time usage events while performing rerating on that customer's account. ECE can also apply top-ups that come in from third-party top-up systems while performing rerating on that customer's account.

A usage request that comes from the network when the customer is in rerating is called a *concurrent request*. Authorization requests and reauthorization requests (for concurrent requests) are processed and the results are returned to the network mediation software program, even when rerating is occurring for that customer.

Concurrent rating applies to ECE usage-event processing. ECE reprocesses any concurrent request received when the customer is in rerating mode. Balances for the customer that have been modified due to BRM rerating are recalculated again in ECE so the balances are synchronized.

The following summarizes the overall process for rerating ECE-rated events in a BRM charging system:

1. In BRM, the system administrator runs **pin_rerate** to create rerating jobs for rerating specified accounts.
2. BRM locks the customer accounts in BRM and sends a prepare-to-rerate message to ECE through the BRM Advanced Queuing (AQ) database queue.

Note: **pin_rerate** sends all rerating messages to ECE by sending business events through the BRM Account Synchronization Data Manager (DM). The rerating messages are converted to update requests by EM Gateway and are sent to ECE through Customer Updater like any other update request. Update requests for rerating messages are referred to as *charging requests*.

3. ECE marks the customers that are part of the prepare-to-rerate request with an in-rerating status and pushes rated events for those customers from the Oracle NoSQL database to BRM so that balances in ECE and BRM are synchronized before rerating starts.
4. ECE uses the Acknowledgment Queue to send an acknowledgment that rerating can be initiated and BRM initiates the rerating process.
5. BRM sends the events to be rerated through EM Gateway to be rated by ECE.
The customer's balances are not affected for the events rated through EM Gateway.
6. BRM sends rerating requests to ECE containing the balance impacts to sponsoring accounts (for charge distribution) as a result of rerating accounts.
7. ECE rerates a customer's usage events according to the rerating requests.

If ECE receives concurrent requests, such as concurrent usage requests or concurrent top-up requests, it rates usage and applies top-ups for the concurrent requests accordingly.

8. When the rerate job is complete in BRM, BRM sends a message to ECE through the BRM AQ database queue.
9. ECE performs necessary balance updates and backouts and synchronizes the balances for customers that were in the rerating requests.

For charge distribution scenarios (charge sharing in BRM), events for customers that were not in the rerating job, but who share discounts or charges with customers who were in the rerating job, are processed further so that balances are applied correctly.

10. After ECE rerates the events, ECE sends the charging results to BRM by generating call details record (CDR) files for Rated Event (RE) Loader to load in the BRM database.
11. ECE generates an acknowledgment that rerating is complete and sends it to BRM through the Acknowledgment Queue.

BRM stores all subscription events that occur in the BRM system until rerating is complete. After rerating, BRM sends the subscription events to ECE as update requests through Customer Updater. Likewise, ECE stores all concurrent-request events that occur in the ECE system until rerating is complete.

If ECE cannot rerate events for a customer, ECE sends a notification to BRM using BRM Gateway. BRM uses the information in the notification to create a rerate job for that customer. If the process of rerating fails during the rerating process, ECE logs messages in the Customer Updater log file.

ECE moves failed rerating requests from Customer Updater to the suspense queue. See the discussion of handling failed update requests from BRM in *ECE System Administrator's Guide* for information about handling failed events in the suspense queue.

About Advice of Charge

ECE supports the 3GPP Advice of Charge (AoC) supplementary service by which customers can be informed about the cost for a requested service either in monetary format or non-monetary format. AoC may be provided at the beginning of a session, during a session, or at the end of a session.

To support AoC, ECE calculates the cost of using a service and relays that information back to the network mediation software program which can then pass the message on to the customer.

For information about configuring AoC in ECE, see the discussion of usage-charging configuration in *BRM Elastic Charging Engine System Administrator's Guide*.

About Advice of Promotion

ECE provides the capability to provide Advice of Promotion (AoP) information enabling the operator to notify the customer that a better price could be obtained for the service they are about to use. ECE provides the AoP information as defined by 3GPP (TS22.086). The operator can choose to use this information in whatever way is relevant to the specific service being offered. For example, a network operator can send the AoP information in an IVR pre-call announcement for a Voice service.

The AoP function can be used to notify a customer of a preferential price that would be available within the configured time window. If a reduced rate is available within this window, then the AoP is returned with an indication of the price to which the new rate would be applicable.

To support AoP, ECE determines whether a better rate for using a service is available near the time that the customer's usage request is received. ECE relays that information back to the network mediation software program, which passes the message on to the customer.

ECE implements AoP as follows:

1. A customer makes a request to initiate a session, to debit a specific or calculated amount of a resource, or to generate a price estimation for using a resource.
2. The ECE charging server calculates the charge for the request.
3. If AoP is enabled, ECE adds a time offset to the start and end time of the request and recalculates the charge using the offset time period (the new start and end time).

You configure how much of a time offset to use. See the discussion of usage-charging configuration in *BRM Elastic Charging Engine System Administrator's Guide* for instructions.

4. If the recalculated charge is less expensive for the customer, ECE sends the information about potential savings back to the network mediation software program in the usage response.

ECE applies AoP when AoP is configured at the ECE system level. Configure AoP at the system level by using the configuration service. See the discussion of usage-charging configuration in *BRM Elastic Charging Engine Implementation Guide* for information about how to configure AoP.

Note the following details about AoP:

- AoP is not configurable in PDC.
- AoP is a systemwide configuration (it is not configured on a per-charge-offer basis).
- By default, AoP gives advice based on time.
- When applying AoP, ECE uses the charge offers and discount offers that are eligible when the request is received to recompute the charge for the offset time period. If a different charge offer or a different discount offer applies to the future offset time period, AoP may advise a promotion when none exists or may not advise a promotion when one is available.

When using AoP, ensure that your rate plans have tiered consumption configured accurately so as to prevent a credit breach of non-currency balance elements.

About Multiple-Service Credit Control

ECE supports multiple-service credit control (MSCC) requests in which a Diameter application performs credit control for multiple services within the same session.

An MSCC request is a list of sub-requests that are targeted to the same customer and share the same operation type and session ID but individually apply to different products.

When ECE receives MSCC requests, it assigns a different session ID to each of its sub-requests. Doing this enables ECE to distinguish one sub-request from another when looking up the active session associated with each sub-request.

An MSCC request results in an MSCC response containing a sub-response for each sub-request. Each sub-response contains status indicating whether the sub-request succeeded or failed.

When ECE saves rated event information for MSCC requests in the Oracle NoSQL database, note the following points:

- Rated event information is saved for each sub-request.
- The NoSQL key for the rated event is based on the session ID that ECE assigned (not based on the original MSCC request session ID).
- The ECE session ID in the Oracle NoSQL database is a composite of the original usage request's session ID, the product type, and the user identity, separated by underscore characters. For example:

Original MSCC request ID: **1313b2ab-d51e-4545-8bba-25c731daf10b**

Usage request's product type: **VOICE**

Usage request's user identity: **650123555**

ECE session ID: **1313b2ab-d51e-4545-8bba-25c731daf10b_VOICE_650123555**

MSCC support applies to usage requests and query requests.

MSCC support does not include support for rating groups (Rating-Group attribute-value pair (AVP)), credit pools (G-S-U-Pool-Reference AVP where units of the service type are pooled in a credit pool) and credit control (as described in section 5.1.2 of IETF RFC 4006).

Refer to the `SampleMultipleServicesLauncher` sample program in the ECE SDK for an example of how to send MSCC requests to ECE. For more information, see the discussion about the ECE sample programs in *BRM Elastic Charging Engine Implementation Guide*.

About Event-Based Charging

For event-based charging, the usage request is for usage that was rendered in a single operation. The usage request can be received before usage, during usage, or after the occurrence of the usage. The data for event-based requests typically maps to the ECE Debit, Refund, Balance_Query, and Price_Enquiry charging operation types.

About Session-Based Charging

For session-based charging, the usage request submitted by the online mediation system are usage reports for a session. The online mediation system uses the START, INTERIM, and STOP accounting data, which maps to the ECE Initiate, Update, and Terminate operation types.

About Midsession Rated Events

To enable BRM to recognize revenue generated during online network sessions, ECE must create rated events and send them to BRM.

By default, ECE generates a rated event for a network session only when the session is ended by a Diameter terminate operation.

Some sessions, however, such as data streaming, last days, weeks, or even months. If you do not want to wait until the end of a lengthy session to recognize revenue for the part of the session that subscribers have already consumed, you can configure ECE to generate a rated event whenever a Diameter update operation occurs during a network session. Such events are called *midsession rated events*.

Midsession rated events enable BRM to recognize revenue incrementally during long network sessions, preventing large amounts of unrated usage and unrecognized revenue from accumulating. They also enable you to show subscribers their running balance throughout a session.

Each midsession rated event:

- Is considered a separate record by downstream processes, such as rerating (treated as individual sessions) and invoicing (treated as individual records on itemized invoices).
- Marks the end of a subsession of the network session. For the subsequent subsession, the network session's volume and duration counters are reset to 0. Therefore, if you enable this feature, do not base your pricing on cumulative volume or duration across a network session.
- Contains a reference to the network session ID and to the update operation in which it was created.

To configure ECE to generate midsession rated events, see the discussion about configuring ECE to generate midsession rated events in *BRM Elastic Charging Engine Implementation Guide*.

About Discounts

ECE supports the following discounting capabilities:

Note: *Discounts* is a BRM term. In ECE, they are typically referred to as *alterations*.

Note: Alterations are always capped to a maximum of one hundred percent of charges.

- Percentage discounts
- Fixed amount discounts
- Tiered discounts
- Threshold-based discounts
- Discount increments (beats)
- Support for parallel (original charge), sequential (remaining charge), and quantity (cascading) discount modes

- Discount priorities

Charge offers and discount offers have a priority assigned to them. When an event is rated, ECE applies the offers in the order of priority.

About Balance Management

ECE stores the current view of the customer balance in the charging system. The active customer balances are updated in real time when usage-request transactions occur in Elastic Charging Server.

ECE publishes rated event information (the charging result of usage-request processing). Other applications in the charging system that also store customer balances use the rated event information to update their customer balances. ECE publishes rated event information to the Oracle NoSQL database.

Customer balance data is kept current by Customer Updater, which synchronizes data from BRM to ECE when activity to the customer account in BRM can impact the correctness of ECE balance data. For example:

- Rating and discounting in ECE result in balance impacts that must be reflected in the BRM database. For example, if a discount results in the reduction of a free minutes balance, that change must be reflected in the BRM database so that the balance can be displayed accurately in Customer Center. These updates are made in the BRM database by RE Loader when it loads rated event CDRs into BRM.
- Activity in the BRM database affects balances used in ECE rating. For example, a customer might purchase a discount that provides 500 free minutes or a customer service representative (CSR) could post a debit to a balance. These balance changes must be reflected in the balance managed in ECE. When these balance changes occur, BRM uses the Account Synchronization DM to send the balance impact to ECE. ECE uses Customer Updater to load the balance impact into ECE caches.

About Charge Distribution

ECE supports *charge distribution* (referred to as charge sharing in BRM).

ECE can support billing systems that enable sharing groups: customers that share discounts or charges with other customers. For example, ECE supports discount sharing groups and charge sharing groups defined in BRM.

Alteration and charge sharing agreements between customers enables discount and charge sharing to be applied in the ECE rating module. ECE supports discount sharing and charge sharing at account level and also at product level. ECE supports only products to participate in an alteration or charge sharing agreement.

About Reservation Validity

ECE authorizes and reserves a balance or *active reservation* for a session request. ECE bases the active reservation on the requested service units of the session request (Initiate or Update request types). ECE sends the validity time for the active reservation or *reservation validity* back to the network mediation client. Reservation validity specifies how long a session can continue before the client must ask for a reauthorization of resources for further usage.

ECE sends a *reservation expiration* back to the network mediation client. Reservation expiration specifies how long a session can continue before the client must report the consumed usage to ECE (report the used service units).

For information about configuring the reservation validity and reservation expiration values that ECE sends back to the network mediation client, see the discussion of configuring charging business rules in *BRM Elastic Charging Engine Implementation Guide*.

About First-Usage Validity

The balance item's first usage in the balance impact sets the validity. When ECE receives a usage request for which a customer first uses a balance item, the validity start time is set to the session start time of the request.

When a first usage balance item's validity period is set, the validity period must be updated in the BRM database also. BRM Gateway uses the notification framework to send the information to BRM. When a first usage validity is initialized, ECE generates the **FirstUsageValidityInitializationEvent** service event. BRM Gateway uses the service event and triggers the opcode on BRM to update the validity of the balance item in the BRM database.

Granting First Usage Resources During Rating

ECE can grant balance impacts during rating that have a validity mode set to first usage. For example, you could create a charge offer that includes these balance impacts:

- Debit 5 cents per minute if there are no included minutes.
- Credit 1 minute for each minute paid at 5 cents per minute. These minutes are valid on first usage.

In this example, the charges occur as follows:

1. A subscriber has used up all his included minutes and is being charged 5 cents per minute.
2. After 10 minutes, the subscriber terminates the call. The subscriber is granted 10 minutes.
3. The next call that the subscriber makes uses the 10 free minutes, granted as first-usage balance impacts.

You can also grant first-usage balance impacts by using a discount. For example, you could create:

- A charge offer that charges 5 cents per minute
- A discount that credits one SMS message for each called minute

In this example, the charges occur as follows:

1. A subscriber makes a 10-minute call.
2. The subscriber terminates the call. The subscriber is granted 10 SMS messages, valid at first usage, with a validity end date after 30 days.

For more information about creating product offerings, see the PDC documentation.

About Reverse Rating

Reverse rating means that ECE uses the customer's credit limit to calculate the amount of usage the customer can afford for a requested service before the service is delivered. For online session-based charging, ECE supports reverse rating. For event-based charging, ECE supports reverse rating for debit requests.

ECE performs reverse rating as follows:

1. ECE receives a usage request of the Initiate, Update, or Debit operation type for a requested resource quantity.
2. ECE calculates the charges for the usage request based on the product offering associated with the customer. A product offering represents the products available to your customers and the price of those products.
3. When the rated result of the charge calculation is applied to the balance of the customer, the impact on the balance exceeds the customer's credit limit for the requested resource quantity.
4. ECE determines the quantity of balance element that can be authorized based on the customer's credit limit and returns the quantity in the usage response to the network mediation software program.

When ECE performs reverse rating for a service in which events are rated by using multiple ratable usage metrics (RUMs), you can configure rounding options. See the discussion of reverse rating when rating is based on multiple RUMs in *BRM Elastic Charging Engine Implementation Guide* for details. ECE can also round up the results of its maximum quantity calculations to the nearest whole number.

About Policy-Driven Charging

ECE supports policy-driven charging. Policy-driven charging implements network, customer, and service policies that can be used by service providers for improving customer experience and for making efficient use of network resources. Service providers can use policies for various reasons, such as to control data usage, set quality of service (QoS), allocate amounts of bandwidth to each service, enforce parental controls, implement charging rules, and so on.

Policies can be service and network aware. Network-aware policies can be created for specific access technologies where the network condition can dynamically alter prices. Service aware policies can be created to provide control over how a customer consumes network resources.

ECE supports policy-driven charging based on PCRF, defined in the 3GPP TS 23.203 v9.9.0 (2011-06) specification.

To support policy-driven charging, ECE exposes the following information in its in-memory data grid to policy clients, such as network mediation software programs. Policy clients use the ECE policy management APIs to retrieve the information and send it to the PCRF:

- Policy label information

Policy enforcement programs on the PCRF use policy labels such as status labels. For example, a QoS label might be defined as **normal-QoS** or **low-QoS** as shown below:

```
<policy_label>
  <label>Basic Subscription</label>
  <resource_code>MBU</resource_name>
  <resource_id>100012</resource_id>
  <unit>megabyte</unit>
  <tiers>
    <tier>
      <range_start>0</range_start>
      <range_end>300</range_end>
      <status_label>normal-QoS</status_label>
    </tier>
  </tiers>
</policy_label>
```

```
</tier>
<tier> <range_start>301</range_start>
      <status_label>low-QoS</status_label>
</tiers>
</policy_label>
</policy_labels>
```

Policy label information is stored in the offer profile object in BRM. ECE extracts this information into its data grid when it extracts customer data from BRM and sends the status label information to the PCRF program when requested.

Note: The offer profile is a reusable pricing object. You create offer profiles in BRM and you associate offer profiles with charge offers and discount offers in PDC using provisioning tags. The label name defined in the offer profile is set as a parameter in the customer account.

- Policy counter information

The Sy interface of the ECE Java policy API transfers policy counter information from ECE to policy clients; it provides policy counter status reporting and policy counter status change notifications.

Policy counters track a customer's usage of a service. For example, ECE tracks how many megabytes of data a subscriber has downloaded. The policy client retrieves the policy counters from ECE and sends them to the PCRF for evaluation.

- Subscriber preferences information

ECE stores subscriber preferences associated with how the customer would like to receive policy notifications. Policy clients can retrieve this data from ECE using the Sp interface (implemented as Sh) of the ECE Java policy API. Subscriber preferences such as the following can be sent to the PCRF:

- Customer's charging-related information (for example, if the customer purchased a Gold, Platinum, or Bronze package)
- Customer's preferred channel for receiving notifications (for example, email or SMS)
- Customer's language

To support policy-driven charging, ECE publishes policy notifications. Offer profiles can store threshold definitions for specific resources. ECE can use the threshold definitions to publish notifications when thresholds are breached (SpendingLimit notifications). Also, when the subscriber preferences of a customer changes, ECE publishes notifications with the new or changed preference information (SubscriberPreference notifications). ECE sends notifications to the JMS notification queue. The policy client, such as Diameter Gateway, listens on the queue and uses the data in the notifications to send Sy and Sp messages to the PCRF.

ECE publishes policy notifications only for products that have active policy sessions. When the policy client initiates policy sessions, it subscribes for receiving the policy notifications on behalf of the PCRF.

When a customer purchases a new product, the PCRF will query the policy label and policy counter (Sy data) again so that it can subscribe for the additional counters associated with the new product.

For more information about policy notifications, see "[About Notifications Used by Policy Clients](#)".

For an overview of how ECE processes policy sessions requests, see "[How ECE Processes Policy Requests for Online Network Mediation Software](#)".

For information about the ECE policy management API, see the discussion about integrating policy clients with ECE in *BRM Elastic Charging Engine Implementation Guide*.

About Subscription Products and Member Products

ECE supports *subscription products* and *member products* (referred to as subscription services and member services in BRM). ECE can rate usage for customers' subscriptions.

See the discussion about managing customer's subscription-level services in the BRM documentation for information about setting up services to track and rate usage and create bills for customers' subscriptions.

Event Charging with Unit Reservation

For online charging, ECE aligns with the Diameter Credit Control Application (DCCA) defined in RFC 4006 in support of event charging with unit reservation (ECUR).

Session Charging with Unit Reservation

For online charging, ECE aligns with the DCCA defined in RFC 4006 in support of session charging with unit reservation (SCUR).

Immediate Event Charging

For online charging, ECE aligns with the DCCA defined in RFC 4006 in support of immediate event charging (IEC). For IEC, usage request are for a direct debit operation. The direct debiting operation may be carried out before the service is delivered.

Server Initiated Reauthorization Requests

ECE supports server-initiated reauthorization requests (RARs).

ECE can perform server-initiated reauthorization during an ongoing session. This enables you to update a session in response to changes that occur to a customer's product offerings or balance (for example, a change to a charge offer or to a Friends and Family promotion). When ECE notifies the network, the network sends a reauthorization request, and, if there is a change in the charge, ECE can base the reauthorization on the new charge.

A server-initiated reauthorization can be triggered from the following conditions:

- Changes to offers, such as the creation, modification, or deletion of a subscriber's charge offer or alteration offer.
- Changes to balances that affect rating (for example, a resource that expires mid-session, a resource that becomes available from a top-up, or changes to the customer balance due to an accounts receivable action).

- Changes to promotions, such as changes to Friends and Family or a Special Day offer.
- Changes to charge sharing or alteration sharing groups. For example, a new member is added to the group or a member is removed mid-session.

For example:

1. A subscriber is in a call session. The subscriber adds the called number of that session to a Friends and Family list.
2. Because a Friends and Family discount might change the charge amount, ECE sends a request to the network.
3. In response, the network sends a reauthorization request.
4. ECE sends a reauthorization, using the Friends and Family charge amount.

Note: A reauthorization request is not triggered by a top-up or by rerating when resources are added to a sharing group owner's account.

To configure server-initiated reauthorization, you enable asynchronous RAR notifications. You also set the **offerEligibilitySelectionMode** variable of the `charging.server` MBean. For more information about configuring server-initiated reauthorization, see the discussion about configuring run-time options in *BRM Elastic Charging Engine Implementation Guide*.

About Taxation Support

ECE supports a fixed-rate tax (a flat-rate taxation also known as GST or VAT).

You can apply a tax on both charges and alterations (discounts).

Tax rates are configurable through ECE business parameter configurations; the tax codes created in BRM should be configured again in ECE. You can use the configuration service to configure taxation parameters. You are required to set mandatory parameters for taxation to work in the ECE run-time environment. See the discussion of usage-charging configuration in *BRM Elastic Charging Engine Implementation Guide* for instructions on configuring taxation.

ECE does not support jurisdiction-based taxation.

About Using Closed User Group Profiles to Determine a Price

ECE can apply pricing based on a subscriber's participation in a closed user group.

The closed user group must be on the same operator network.

Operators can set up closed user groups at the customer level or at the product level or both. Customers become a member of a closed user group when they purchase an offer that has a closed-user-group profile associated to it.

Special rates for closed user groups are determined through a custom rule and generic selector that you configure in charge offers. When ECE receives a request to initiate a calling session between two members of the same closed user group, ECE evaluates the charges based on the generic selector's custom-rule configuration and determines the price accordingly.

When the called and the calling customers have more than one closed user group in common, ECE applies the same closed-user-group pricing (the same rate or discount) regardless of the group. To configure different rates for different groups, you can use ECE pre-rating extensions. The customer profile data of both the calling customer and the called customer is accessible to the ECE pre-rating extension.

For information about setting up a custom rule for closed-user-group calls, see the discussion about configuring closed user groups in *PDC User's Guide*.

For information about verifying that closed-user-group calls are processed by ECE during testing, see the discussion about testing charging scenarios in *BRM Elastic Charging Engine Implementation Guide*.

About Assigning Balance Impacts to Bill Items

ECE can assign balance impacts for the same event to different bill items based on rules you configure.

By configuring how balance impacts are assigned to bill items, operators have flexibility in classifying usage into different bill item categories (or buckets). For example, if an operator defines custom bill items for international and national calls, the usage for each type of call can be accounted for as required in separate bill items.

To assign balance impacts to different bill items, ECE evaluates an item type selector rule at run time to derive which item types to assign balance impacts to. The item type selector rule is an expression that can reference any of, or a combination of, the following attributes:

- Event attributes
- Attributes of the charging result

Item type selector rules are contained in item type selectors. For information about setting up item type selectors, see the discussion about configuring item type selectors in *PDC User's Guide*.

About Processing Delayed Usage Requests

Typically, ECE receives usage requests during the same accounting cycle in which the usage occurred. However, some usage requests are received after the end of the accounting cycle in which the usage occurred. These requests are called *delayed usage requests*.

If you configured delayed billing in BRM, you can configure ECE to process delayed usage requests for the accounting cycle in which the usage occurred if the usage requests are received within the delay tolerance interval that you specify. This extends the item assignment for that accounting cycle by the specified interval so that the associated rated events are assigned to the current open bill item. Any delayed usage request received after the specified interval is processed only for the next accounting cycle and the associated rated event is assigned to the next open bill item.

To decide which accounting cycle the delayed usage request belongs to, ECE considers the following dates:

- The date when the usage occurred.
- The current system date.
- The date when the current accounting cycle ends. This is called the *next accounting cycle date*.

- The delay tolerance interval, which is the number of days after the current accounting cycle ends during which delayed usage requests are processed for the current accounting cycle. This interval must be less than the delayed billing interval (the value of the `config_billing_delay` entry in the `BRM_home/sys/cm/pin.conf` file).

For example, consider the following dates:

- The usage occurred on October 23.
- The current system date is October 26.
- The next accounting cycle date is October 25.
- The delay tolerance interval is 4 days (October 25 through October 28).

In this case, if the usage request is received on October 26, the usage request is processed for the current accounting cycle because it is received within the delay tolerance interval. If the usage request is received on or after October 29, which is after the delay tolerance interval, the usage request is assigned to the next accounting cycle.

To enable ECE to process delayed usage requests for the accounting cycle in which the corresponding usage occurred, see the discussion about configuring item assignment in ECE for BRM in *BRM Elastic Charging Engine Implementation Guide*.

About Redirecting Subscriber Sessions to a Service Portal

Service providers can redirect a subscriber session to a service portal, a server outside of the online charging system, where specific services can be offered to the subscriber. ECE can be configured to send service portal addresses back to credit-control clients in its usage response. Credit-control clients use the information for redirecting a session (often referred to as *redirection*) to the service portal applicable to the business scenario.

Redirection is used primarily when the subscriber no longer has a balance available to continue the session without a top-up. The redirection can occur at the last leg of an ongoing session where no more balance is available or at any subsequent requests for which no balance is available (for example, the account is in a recharge-only state). Redirection is also used for user scenarios where the subscriber has an inactive account.

When the credit-control client receives the Final-Unit-Indication in the CCA answer from ECE, the credit-control client behavior depends on the value, TERMINATE or REDIRECT, indicated in the Final-Unit-Action AVP. If you do not configure redirection rules in ECE, then ECE indicates a Final-Unit-Action of TERMINATE in the usage response.

For information about configuring ECE for redirecting a session to a service portal, see the discussion about configuring business rules for charging in *BRM Elastic Charging Engine Implementation Guide*.

About Usage Requests

This chapter describes how client programs create and submit usage requests to Oracle Communications Billing and Revenue Management (BRM) Elastic Charging Engine (ECE).

Usage Request Overview

ECE can process network charging requests from any usage domain (telco, toll, content, and so on). Network mediation software programs map usage attributes that are relevant to charging from the network charging requests to usage attributes in ECE usage requests.

ECE usage requests contain both fixed (mandatory) attributes and a payload that holds dynamic (configurable) attributes to represent the usage of the customer.

See the usage request Java API in *BRM Elastic Charging Engine Java API Reference* for a detailed description of fixed attributes and the payload.

Client programs use the Elastic Charging Client API provided in the ECE SDK for building and submitting usage requests to the Elastic Charging Server.

Client programs can choose between incremental or cumulative accounting behavior when instantiating the usage request builder. See the discussion of implementation concepts in *BRM Elastic Charging Engine Implementation Guide* for more information.

After network mediation software programs create usage requests, they submit them to the Elastic Charging Server (Elastic Charging Server instances are referred to as ECE charging server nodes). Refer to *BRM Elastic Charging Engine Java API Reference* for information about the methods network mediation software programs use to submit usage requests. Refer to the ECE sample programs for examples of how usage requests are built and submitted to the Elastic Charging Server.

Network mediation software programs create usage requests when the charging requests (online) or CDRs (offline) are received from the network. The programs call the ECE API to construct and submit the usage request. The programs read the usage response returned by the Elastic Charging Server after the Elastic Charging Server processes the usage request.

The Elastic Charging Server reads the submitted usage request, processes it, and sends a usage response. The network mediation program calls the ECE API to read the response. See "[About Usage Responses](#)" for more information.

How Network Mediation Systems Create and Submit Usage Requests

Online and offline mediation systems transform charging requests (online charging) and CDRs (offline charging) that are received from the network into ECE *usage requests*.

A program calls the ECE API to construct and submit the usage request. For ECE to recognize and be able to process the chargeable events within the charging requests and CDRs, the usage request adheres to a format that serves to map attributes from the network to attribute definitions described in the ECE DSL language. Usage requests contain both fixed attributes and dynamic attributes. See the discussion about usage request fixed attributes in *BRM Elastic Charging Engine Implementation Guide* for more information.

ECE reads the submitted usage request, caches it, and sends a usage response. The network mediation software program calls the ECE API to read the response. See "[About Usage Requests](#)" for more information.

The attributes from the network charging requests or CDRs which are relevant for charging are mapped to the fields in the event definition. These attributes or *payloads* are used by ECE for rating.

After the usage request is created, the network mediation software program uses the **brs.submit()** method to submit the usage request to the Elastic Charging Server.

About Usage Responses

This chapter describes the usage responses Oracle Communications Billing and Revenue Management (BRM) Elastic Charging Engine (ECE) charging servers can return after processing usage requests.

Understanding Usage Responses

After ECE processes a *usage request*, ECE sends a *usage response* to the system that sent the usage request.

ECE returns usage responses to the network mediation system after processing usage requests. Usage responses include reason codes for indicating how the request was processed in the system. For telecommunications, the usage response is sent back to the network mediation system (or to Diameter Gateway) after charge calculations have been made and contains the result of the ChargingRequest operation.

For offline charging, ECE can send a negative usage response back to the network mediation system when the CDR related to the usage request must be suspended. A negative usage response can occur if the customer has not yet been provisioned in ECE, or if the customer does not have a pricing offer that can rate the request.

For online charging, ECE can send a usage response that contains an *in-session notification* about the active customer session. The in-session notification is a block within the usage response. For example, ECE can send an in-session notification about a customer's eligibility for a better rate during a calling period (in support of the Advice of Promotion feature) or about a customer's threshold breach information if they breached one of their credit limits for a particular service. ECE can also send the customer's remaining-balance information within the usage response. The network mediation system can use the data from the in-session notification for sending a message to the customer. See "[About Advice of Promotion](#)" for information about the data ECE sends for Advice of Promotion feature. See "[About Notifications](#)" for more information about how you can configure the ECE notification framework to include in-session notifications in the usage response.

For information about the different reason codes ECE returns in the usage response, refer to the documentation for `oracle.communication.brm.charging.messages.usage` in *BRM Elastic Charging Engine Java API Reference*.

About Notifications

The Oracle Communications Billing and Revenue Management (BRM) Elastic Charging Engine (ECE) notification framework publishes notification events which can be used by external systems. For example, for policy-driven charging, policy-related software programs (*policy clients*), can use the data in notification events for sending information to the Policy and Charging Rules Function (PCRF). This chapter describes the purpose of publishing notification events and how the notification framework publishes them.

See the discussion of configuring notifications in *BRM Elastic Charging Engine Implementation Guide* for information about configuring the service events that trigger the publishing of notification events.

Overview of Notifications in ECE

ECE supports *in-session* notifications and external asynchronous notifications. See the following topics for information about each notification type:

- [About In-Session Notifications](#)
- [About External Notifications](#)

You can enable or disable ECE from publishing external notifications for various events that occur in the ECE system. You can enable or disable ECE from sending in-session notifications as part of the usage response. See the discussion of configuring notifications in *BRM Elastic Charging Engine Implementation Guide* for more information.

About In-Session Notifications

In-session notifications are returned in the usage response to the network mediation system. The online mediation session controller uses the in-session notification information and the customer preferences to determine the messages that must be sent to the customer.

ECE supports in-session notifications for the Advice of Charge (AoC) charging scenario in addition to supporting asynchronous external notifications for AoC. AoC notifications include the operation types to indicate when the notification is triggered. For USAGE operations, the AoC notifications include the subtype. See "[About Advice of Charge](#)" for information about AoC. ECE also supports in-session notifications for threshold breach scenarios when a customer's balance breaches a credit threshold value.

For details about in-session notifications returned in the usage response for AoC, refer to the documentation for **oracle.communication.brm.charging.messages.usage** in *BRM Elastic Charging Engine Java API Reference*.

ECE supports in-session notifications for policy-driven charging by publishing asynchronous external notifications during a policy session. Policy clients consume the data in these notifications for sending in-session notifications to the PCRF.

For details about in-session notifications returned in the policy response for policy-driven charging, refer to the documentation for **oracle.communication.brm.charging.messages.policy** in *BRM Elastic Charging Engine Java API Reference*.

About External Notifications

When certain events or state changes occur in ECE, ECE can publish asynchronous external notifications to a JMS queue. The external notifications contain the information about the event. For example, when a customer breaches one of his credit thresholds, ECE publishes the `ThresholdBreachServiceEvent` which contains the information about the breach such as the balance element (currency or non-currency) and the quantity of the breach (amount or percentage).

The external notification payload that is published into the JMS queue has a well defined XML format. External systems parse the XML strings of the payload to obtain the required information such as the balance element and current balance and so on. External notifications are published as XML strings in the XML structure shown by the `ECE_Home/occeserver/config/Notification.xsd` file, where `ECE_Home` is the directory in which you installed ECE.

For samples of XML strings published for different notification types, see the discussion of configuring notifications in *BRM Elastic Charging Engine Implementation Guide*.

Examples of client applications that would obtain information from external notifications for their own processing are:

- The network mediation system can use data in the external notification with customer policy data for implementing network policy control.
- Oracle Communications Billing and Revenue Management (BRM) can use data in the external notification for running billing for a specific customer. BRM Gateway sends the relevant data to BRM in the external notification for triggering billing.
- BRM can use data in the external notification for updating the first usage validity period of a customer's balance item. BRM Gateway synchronizes the first-usage validity settings from ECE to BRM when first-usage notifications are configured.

Specific ECE service events trigger ECE to publish external notifications. You can enable or disable each of the service events from triggering external notifications. See ["About Service Events that Trigger External Notifications"](#) for information about the service events for which ECE supports publishing external notifications.

You must configure JMS credentials for publishing external notifications. See the discussion of configuring notifications in *BRM Elastic Charging Engine Implementation Guide* for instructions on configuring the JMS credentials in ECE.

See the discussion of configuring notifications in *BRM Elastic Charging Engine Implementation Guide* for information about enabling and disabling external notifications.

External notifications are published by the notification framework. See "[About the Notification Framework](#)" for information about configuring ECE to publish external notifications for various service events.

About Notifications Used by BRM

ECE sends notifications to BRM for synchronizing data and triggering required processes such as billing. For more information, see the discussion of implementing ECE with BRM in *BRM Elastic Charging Engine Implementation Guide*.

The following notifications are typically enabled as a best practice when integrating with the BRM system:

- Billing notifications

Trigger billing in BRM for a specific customer (BillingServiceEvent).

When a subscriber starts a charging session near the time billing is set to run in BRM, ECE creates the BillingNotificationServiceEvent service event. If notifications are enabled for this service event, when this event occurs, the notification framework publishes the BillingNotificationServiceEvent notification to the JMS notification queue.

When a BillingNotificationServiceEvent notification is published to the JMS notification queue, BRM Gateway, which listens on the queue, uses the data from the event to create a BRM request. The BRM request calls an opcode in BRM to trigger billing in BRM for the customer.

Triggering billing for a customer when the customer starts a charging session near the time billing is set to run enables BRM to apply new cycle grants to the customer balance and to process any discounts or folds applied during billing. BRM then sends the information to ECE, which updates the customer balance in ECE. In this way, resources associated with these operations are replenished in ECE, which decreases the risk of terminating the charging session for lack of resources if resources were low in the account when the charging session began. It is a best practice to enable notifications for this service event (setting to ASYNCHRONOUS).

For offline charging, when a subscriber starts a charging session during the delayed billing period, ECE rejects the request for charging and sends a notification to BRM to trigger the partial billing. If new billing notifications are triggered for the same accounting cycle, ECE detects duplicate notifications and sends the notifications only if they are not sent earlier for the same accounting cycle.

External notifications are published as XML strings in the XML structure shown by the *ECE_home/occeserver/config/Notification.xsd* file.

- Replenish POID notifications

To track a customer's balance impacts for a given billable item type, ECE maps the product type and event type combinations in a usage request to the relevant item type. ECE sends information to BRM so that it can obtain the POIDs when it does not have enough that can be used (ReplenishPoidIDNotificationEvent).

- Life Cycle Transition notifications

When the life cycle state changes in ECE, this notification is used to synchronize the state with BRM. If life cycle management is enabled in ECE, this notifies BRM of any changes in the subscriber life cycle state (LifeCycleTransitionServiceEvent).

- First Usage Validity Initialization notifications
Synchronize validity of first usage resources from ECE to BRM (FirstUsageValidityInitServiceEvent).
- External Top Up notifications
Update the balances in ECE and send notifications to BRM (ExternalTopUpNotificationEvent).

About Notifications Used by Policy Clients

When changes occur to policy counters (balances) or policy-related subscriber preferences that are associated with products that have an active policy session, ECE publishes asynchronous notifications to the JMS notification queue. A policy client interacting with ECE subscribes for receiving the policy notifications when it initiates the policy sessions. The policy client uses the information in the notifications to send Sy and Sp messages to the PCRF.

ECE publishes policy notifications for products that have an *active* Sy session if the balance threshold is breached because of a usage session, or a balance top-up, or a subscription activity. ECE does *not* generate notifications for threshold breaches that occur *outside* of an Sy session (when the customer is not using the product) as these breaches could generate unnecessary notifications. For example, it would be a waste of system resources for ECE to publish notifications for breaches caused by monthly resets of resource allowances for products that are not being used by customers.

For Java API descriptions of the service event objects published as policy notifications, refer to the documentation for **oracle.communication.brm.charging.messages.policy** in *BRM Elastic Charging Engine Java API Reference*.

For information about the XML data structure of ECE policy notifications, see the discussion about integrating ECE with policy clients in *BRM Elastic Charging Engine Implementation Guide*.

Spending Limit Notifications

ECE publishes SpendingLimit notifications when it detects a policy threshold breach. ECE generates a SpendingLimit notification when usage results in a threshold breach of pre-defined policy tiers. A threshold breach causes ECE to send a label tagging this condition to the PCRF. Based on this label, the PCRF will take a policy action (such as request the PCEF to throttle bandwidth, or block a service, and so on).

The customer balance change could occur because of any of the following:

- Usage requests coming from the network mediation system
- Update requests coming from BRM (as the customer management system)
- Top ups coming from top-up systems

ECE checks for spending limit breaches and generates notifications for any breaches for products that have an active policy session (Sy session).

ECE administers threshold checks during the usage processing flow (in-session) and during the balance update flow (off-session) based on the current balance and consumed reservation of the customer balance.

Subscriber Preference Notifications

ECE publishes SubscriberPreference notifications during an update from BRM for the customer preferences present in a particular product. For example, if a CSR adds a preference to a customer's profile, modifies an existing preference, or adds a new type of customer preference. ECE publishes SubscriberPreference notifications only for active subscriptions of Sp data (SubscribeNotificationRequest).

About Service Events that Trigger External Notifications

When certain state changes occur in the ECE system, service events are created. When configured, the service events can trigger the publishing of notification events (asynchronous external notifications). See the discussion of configuring notifications in *BRM Elastic Charging Engine Implementation Guide*.

When the following service events occur, they can trigger the notification framework to publish associated external notifications to a JMS notification queue.

- AdviceOfChargeServiceEvent
See "[About Advice of Charge](#)" for information.
- FirstUsageValidityServiceEvent
See "[About First-Usage Validity](#)" for information.
- ThresholdBreachServiceEvent
See "[About Breach-Type Notifications](#)" for information.
- TopUpNotificationServiceEvent
See "[About Top-Up Notifications](#)" for information.
- BillingNotificationServiceEvent
See "[About Billing Notifications](#)" for information.
- ReplenishPoidId
See "[About Notifications Used by BRM](#)" for information.

About Top-Up Notifications

ECE creates top-up notifications in support of Server Initiated Re-auth Requests (RAR). See "[Server Initiated Reauthorization Requests](#)" for details on how ECE can support RAR.

See the discussion of configuring notifications in *BRM Elastic Charging Engine Implementation Guide* for information about configuring this notification.

About Billing Notifications

When a customer starts a charging session near the time billing is set to run in BRM, ECE publishes the BillingNotificationServiceEvent service event. If notifications are enabled for this event, when this events occurs, the notification framework publishes the BillingNotificationServiceEvent notification to the JMS notification queue.

See the discussion of configuring notifications in *BRM Elastic Charging Engine Implementation Guide* for information about configuring this notification.

About Breach-Type Notifications

When performing usage-charging, ECE can generate notifications related to breaches in credit limits and thresholds. A client application can use this information for sending a notification back to the customer according to the customer's user preferences. For example, an email or an SMS could be sent to inform the customer that a credit threshold has been breached.

If notifications are configured for breaches in credit limits and thresholds, when the event occurs, the notification framework publishes a notification for each type of event to the JMS notification queue.

See the discussion of configuring notifications in *BRM Elastic Charging Engine Implementation Guide* for information about configuring notifications.

For information about policy-driven threshold notifications, see "[About Notifications Used by Policy Clients](#)".

About Threshold Breach Notifications

When a customer's balance breaches a credit threshold value, ECE publishes the `ThresholdBreachServiceEvent` notification with information about the breach such as the threshold amount or threshold percentage that was breached.

About Credit Ceiling Breach Notifications

When a customer's balance breaches a credit ceiling, ECE publishes the `CreditCeilingBreachServiceEvent` notification with information about the breach such as the balance element code and balance identifier corresponding to the breach.

About Credit Floor Breach Notifications

When a customer's balance breaches a credit floor, ECE publishes the `CreditFloorBreachServiceEvent` notification with information about the breach such as the balance element code and balance identifier corresponding to the breach.

About the Notification Framework

The notification framework publishes in-session notifications and asynchronous external notifications. When certain state changes occur in the ECE system, service events are created by the Elastic Charging Server. The notification framework, based on how you configure ECE, puts the data from these events either in the usage response (for an in-session notification), or into a notification event (for an external notification), or does both.

The notification framework publishes external notifications asynchronously. To publish external notifications, the notification framework transforms the Java payload of ECE service events into notification events that have a well defined XML payload. These notification events can be consumed by other systems for further processing.

The notification framework publishes external notifications to a JMS topic on a remote JMS server. You must set up a notification topic on a remote JMS server and configure the JMS credentials in ECE to access that server. See the discussion of configuring notifications in *BRM Elastic Charging Engine Implementation Guide* for instructions on configuring the JMS credentials in ECE.

The notification framework is not used to receive messages from external systems.

Transforming the Service Event Java Payload into XML

The notification framework transforms the Java payload of the ECE service event into XML. The payloads that are published into the JMS notification queue (JMS topic) are XML strings in the XML structure shown by the *ECE_Home/occeserver/config/Notification.xsd* file.

For samples of XML strings published into the JSM notification queue for different notification types, see the discussion of configuring notifications in *BRM Elastic Charging Engine Implementation Guide*.

Specifications and Standards Compliance in ECE

This appendix describes the specifications and standards that are used in Oracle Communications Billing and Revenue Management (BRM) Elastic Charging Engine (ECE).

About Specifications and Standards Compliance

The ECE charging API aligns with the Remote Authentication Dial In User Service (RADIUS) Accounting Request for Comments (RFC) specifications and with the standards described in the 3rd Generation Partnership Project (3GPP) Technical Specifications (TS). ECE charging supports any 3GPP sub-domain; some are listed here as examples:

- PS (Packet Switched) connections
- CS (Circuit Switched) connections
- WLAN (Wireless Local Area Network)
- IMS (IP-Multimedia Subsystem)
- PCRF (Policy and Charging Rules Function) and Sy/Sp (Sh) interfaces

The ECE charging API is extensible; it can accommodate proprietary extensions of the standards.

The ECE Java API aligns with the Diameter Ro, Diameter CCA, Diameter Rf, and RADIUS message formats. Network mediation software programs (client applications) that support these protocols can send usage requests to ECE.

The following 3GPP Technical Specifications (TS) relate to ECE charging functionality.

- "3GPP TS 32.240 Telecommunication management; Charging management; Charging architecture and principles" at:

<http://www.3gpp.org/ftp/Specs/html-info/32240.htm>

For online charging, ECE exposes a Java API based on Diameter Ro, which is extensible for supporting any extension or variation. Online mediation systems use this interface to send usage requests to ECE. The online mediation system processes the Diameter requests and maps the content of the Diameter requests into ECE usage requests.

ECE implements the following functionality for online charging as described in 3GPP TS 32.240:

- Online Charging Function modules:

- * Session Based Charging Function (SBCF)
- * Event Based Charging Function (EBCF)
- Rating Function (RF)
- Account Balance Management Function (ABMF)
- "3GPP TS 32.260 Telecommunication management; Charging management; IP Multimedia Subsystem (IMS) charging":
<http://www.3gpp.org/ftp/Specs/html-info/32260.htm>
- "3GPP TS 32.299 Telecommunication management; Charging management; Diameter charging applications":
<http://www.3gpp.org/ftp/Specs/html-info/32299.htm>

For offline charging, ECE exposes a Java API based on DIAMETER Rf, which can be called from the offline mediation system. The Java interface has functionality close to that of the Rf interface described in 3GPP 32.299 and is extensible for supporting any extension or variation.

Oracle Communications Offline Mediation Controller uses this interface to load CDRs into ECE for charging.

- GB922 TM Forum Information Framework (SID) documents can be downloaded from the following location:
<https://www.tmforum.org/resources/suite/gb922-information-framework-r14-5-1-pdf/>

The following RADIUS RFCs relate to ECE charging functionality.

- RFC 2865, "Remote Authentication Dial In User Service (RADIUS)," June 2000, RADIUS. Updated by: RFC 2868, RFC 3575, RFC 5080.
- RFC 2866, RFC 2867, RFC 2868, RFC 2869, RFC 3579

ECE aligns with the Diameter Credit-Control Application charging functionality described in Internet Engineering Task Force (IETF) Network Working Group RFC 4006.

The following 3GPP Technical Specifications (TS) relate to the Policy and Charging Rules Function (PCRF) and ECE.

- "3GPP TS 29.219 Policy and charging control: Spending limit reporting over Sy reference point":
<http://www.3gpp.org/ftp/Specs/html-info/29219.htm>

The Sy interface is located between the PCRF and Online Network Mediation Controller. It enables the transfer of customer spending information.

- "3GPP TS 29.329 Sh interface based on the Diameter protocol":
<http://www.3gpp.org/ftp/Specs/html-info/29329.htm>

The Sp (implemented as Sh) interface is located between the SPR (Subscription Profile Repository) and PCRF. It enables the retrieval of customer identities and profile information.

Glossary

advice of charge (AoC)

A supplementary service that informs customers about the cost of a requested service in monetary or nonmonetary format.

advice of promotion (AoP)

An online charging function by which you can inform customers of a possible promotion if they request a service at a future time; for example, when a call would charge less in an off-peak time period.

BRM Gateway

An ECE component that enables ECE to send data to the BRM server for various purposes; for example, triggering billing, and updating subscriber life-cycle states.

challenge handshake authentication protocol (CHAP)

An authentication protocol that authenticates a user to a network entity; for example, the Web. This protocol ensures that the server sends a challenge to the client after the client establishes a network connection to access the Web server.

charging node

A Java Virtual Machine (JVM) running Coherence that is a member of the Coherence cluster. There can be multiple nodes in a Coherence cluster. A client application that becomes a member of the Coherence cluster becomes a node in the cluster. The [Elastic Charging Controller](#) monitors and manages all nodes the are defined in the ECE topology file.

Coherence cluster

The data grid on a physical machine of which ECE nodes are a member. Client applications that need to send requests to ECE must join the Coherence cluster to access the data in ECE caches and to access other services required for sending requests. Client applications use the Elastic Charging Client API for joining the Coherence cluster.

Customer Updater

An ECE component that keeps the customer data in ECE synchronized with the customer data in BRM.

When changes to customer data occur in BRM, Customer Updater automatically updates the ECE customer cache. Customer Updater updates the cache by adding, modifying or removing data as required by the update.

Customer Updater is a component that listens on a queue for update requests from BRM. Customer Updater is separate from the **customerLoader** which is a utility for loading customer data into ECE from file.

customerLoader

A utility used to load customer data into ECE from file. The **customerLoader** utility loads data that has been previously extracted from BRM by Oracle Data Integrator. **customerLoader** loads the data from the ECE customer XML data files into the customer cache of the Elastic Charging Server.

delayed usage request

A **usage request** that is received after the end of the accounting cycle in which the usage occurred. For example, if the current accounting cycle ends on October 25, the usage request received after October 25 for the usage that occurred before October 25 (assume October 23) is called the delayed usage request.

Diameter Gateway

An ECE component that serves as the Online Charging System (OCS) front-end server for the BRM system. It processes network requests for Gy, Sy, and Sh Diameter interfaces. It acts as a Diameter Server and presents the BRM Elastic Charging Engine to the network as a Diameter Credit-Control Application. It translates Diameter requests received from Diameter Clients (for example, Application Servers, Policy Servers or IMS-GWFs) into Elastic Charging Engine Java API requests. It translates the response from the Elastic Charging Server back into Diameter requests and responds back to the requesting Diameter Client.

driver machine

The driver machine is the machine on which you installed ECE, and the machine used to administer the ECE system. The driver machine is the server that is set up to be the primary administrator machine. As a best practice, you do not install ECE nodes on the driver machine; you install them on separate remote machines (*server machines*).

Elastic Charging Client

The client library that enables client applications to connect to the cluster and build usage requests for sending to the Elastic Charging Server. The Elastic Charging Client is used for sending all types of requests: usage requests, query requests, update requests, and management requests. When client applications start the Elastic Charging Client, they automatically join the cluster. As a member of the cluster, applications have access to ECE node data caches and the services required for sending requests.

Elastic Charging Controller

The ECE command-line application used for operational management of ECE nodes in the cluster.

Extensible Authentication Protocol (EAP)

An authentication protocol which supports multiple authentication mechanisms used for RADIUS; for example, EAP-Tunneled Transport Layer Security.

External Manager Gateway

The ECE component that converts a BRM opcode input list into an ECE request and converts the subsequent ECE response back into a BRM opcode output list. External Manager (EM) Gateway acts as a client to the Elastic Charging Server, sending

requests to it from BRM. For example, EM Gateway is used to send rerating requests and balance query requests from BRM to ECE.

identity cache

The identity cache stores the public user identity information of customers; for example, the customer's phone number, International Mobile Subscriber Identity (IMSI), and Session Initiation Protocol (SIP) address.

JConsole

The JConsole graphical user interface is a monitoring tool that complies to the Java Management Extensions (JMX) specification. You can use JConsole and other JMX editors to edit ECE MBeans.

JMX-management-enabled node

An ECE node that has a JMX port specified and has the **start CohMgt** parameter set to **true** in the ECE topology file. When this node starts, it provides a JMX management service on the specified host and port, which is used to expose ECE configuration MBeans. This enables you to edit the MBean attributes by using a JMX editor such as JConsole. You designate only one node as a JMX-management-enabled node for each unique IP address in your topology.

midsession rated event

A rated event generated when a Diameter update operation occurs during a network session. By default, ECE generates rated events for a network session only when a Diameter terminate operation ends the session. To avoid waiting until the end of a lengthy session to recognize revenue for the part of the session that subscribers have already consumed, however, you can configure ECE to generate midsession rated events each time a Diameter update operation occurs in an ongoing session.

network access server (NAS)

See [terminal server](#).

notification framework

An ECE framework that publishes asynchronous notifications to a database queue when certain events, or state changes, occur in the ECE charging server. For example, when a customer breaches a credit threshold, the details of the threshold breach are published. Other applications can use the information in the notifications for their own processing.

Oracle Data Integrator (ODI)

ODI is used to generate ECE customer XML data files. ODI extracts customer data from the BRM database and transforms it into ECE XML data format. Extracted BRM customer data is loaded into ECE by the **customerLoader** utility.

Oracle NoSQL Database

ECE uses Oracle NoSQL Database to temporarily store rated event information. The ECE components that interact with Oracle NoSQL Database are Rated Event Publisher and Rated Event Formatter. Rated Event Publisher publishes data to the NoSQL data store. Rated Event Formatter formats rated event information for other applications to use; it also deletes the rated events from the NoSQL database once those rated events are processed.

payload

The fields in the request specification that are relevant to usage charging. The fields are used to create rate plan expressions which ECE uses to process usage charging operations.

password authentication protocol (PAP)

An authentication protocol that uses the user name and password to validate users.

Pricing Updater

A process that loads pricing data from Pricing Design Center and updates the ECE pricing cache when alterations to pricing data are published from Pricing Design Center.

pricingLoader

A utility that loads pricing data from an XML file into ECE. **pricingLoader** is only used on test or standalone installations on which Oracle Communications Pricing Design Center (PDC) is not used.

Pricing Updater loads pricing data into ECE automatically when you publish pricing data in PDC for ECE. Do *not* mix running **pricingLoader** and Pricing Updater on the same ECE deployment.

RADIUS Gateway

An ECE component that serves as the OCS front-end server for the BRM system that processes authentication and accounting requests from RADIUS clients. It acts as a RADIUS server and presents BRM Elastic Charging Engine to the network as a RADIUS application. It translates RADIUS requests received from RADIUS clients into ECE Java API requests. It translates the response from Elastic Charging Server back into RADIUS requests and responds back to the requesting RADIUS client.

Rated Event Formatter

An ECE process that reads rated event information from the Oracle NoSQL Database and formats the information as required by other applications that need to use the information. It also deletes the rated events from the NoSQL Database once the events are processed.

Rated Event Publisher

An ECE process that publishes rated event information to the Oracle NoSQL Database.

Remote Authentication Dial In User Service (RADIUS)

An industry standard protocol for processing authentication, authorization, and accounting (AAA) requests. Terminal servers use the RADIUS protocol to communicate AAA requests to, and return results from, a database of customer information. ECE implements RADIUS protocols for authentication and accounting by using RADIUS Gateway.

request specification data

Data that defines the usage payload (rating attributes) that ECE can accept when processing usage requests. The payloads are fields that are relevant to usage charging. The fields are used to create rate plan expressions which ECE uses to process usage charging operations. The payloads and network attributes make up the event definition of your events. You perform event definition in PDC. When you publish your event definition to ECE (from PDC by way of Pricing Updater), the request specification data is automatically generated and published to ECE.

simulator

The ECE tool used to emulate the role of a network mediation software program (client application) sending requests to ECE.

state machine

The application that monitors all states of ECE.

terminal server

A server that your customers connect to when dialing in to your IP services. Also known as Network Access Server (NAS).

topology file

The file (*ECE_Home/occeserver/config/eceTopology.conf*, where *ECE_Home* is the directory in which you installed ECE) that describes all ECE nodes in the Coherence cluster. The topology file defines the name, role, physical host, and JVM tuning file preference for each node. Elastic Charging Controller (ECC) reads your topology file to know where all of the nodes of the cluster are located.

usage request

A request in ECE-domain-specific-language format sent to ECE from client applications, such as network mediation software programs, for charging usage of a product by a customer.

Client applications construct usage requests and send them to ECE. For example, a network mediation software program sends usage requests to ECE so that online or offline charging of network usage can be performed. Usage requests contain payloads ECE requires for calculating charges such as the duration of the call, the data volume used, and so on. Usage requests adhere to the format defined by the ECE [request specification data](#).

usage response

A response by the Elastic Charging Server to the client application that sent a usage request to ECE.

